INFINITESIMAL PERTURBATION ANALYSIS FOR ACTIVE QUEUE MANAGEMENT

A Thesis
Presented to
The Academic Faculty

by

Richelle Adams

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
December 2007

INFINITESIMAL PERTURBATION ANALYSIS FOR ACTIVE QUEUE MANAGEMENT

Approved by:

George Riley, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Yorai Wardi, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

John Copeland
School of Electrical and Computer Engineering
Georgia Institute of Technology

Henry Owen
School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: 31 October 2007

To God

ACKNOWLEDGEMENTS

There have been many people who contributed to my positive sojourn here at the Georgia Institute of Technology, and in Atlanta as a whole. To list them all here and to mention the ways they have aided my experience would be impossible. But to all these persons, I express my heartfelt gratitude to you. Nevertheless, there are some people that I must indeed mention, for without them I would not have seen the ending or even the beginning of this tunnel.

Special thanks to my advisors, Dr. George Riley and Dr. Yorai Wardi, for taking up the challenge of having me as their student. Thank you for helping me carve out a thesis topic. Thank you for your ready accessibility and for the numerous discussions (impromptu and otherwise) on my theoretical, simulation and coding questions. I am grateful for your support and guidance. I thank also Dr. John Copeland and Dr. Henry Owen for agreeing to be part of the Reading Committee and Dr. Monson Hayes and Dr. Jun Xu for agreeing to be part of the Dissertation Defense Committee. To the School of Electrical and Computer Engineering, Georgia Tech, thank you for, among other things, the financial support you provided me through assistantships and teaching opportunities.

I am deeply appreciative to the Public Affairs Section of the U.S. Embassy in Port-of-Spain, Trinidad and Tobago and LASPAU (Academic and Professional Programs for the Americas) who were instrumental in providing me this privilege of pursuing the doctoral degree in the USA. Your belief in my potential to accomplish this was humbling. Thank you for the grant through the Fulbright Faculty Development Program. Special thanks to all the placement staff and advising staff of LASPAU that made the transition into the USA bearable. Thank you for your patience and persistence.

I wish to thank Dr. Brian Copeland, Head of the Department of Electrical and Computer Engineering (DECE), the University of the West Indies (St. Augustine Campus), for encouraging me, just a junior faculty member, to further my studies here in the USA. Thank

you for supporting me with approvals of leave and extensions of leave, and other administrative details, without which I would not have reached to this point. To all the DECE faculty and staff, thank you for the support and patience.

To my precious mother, Theresa Henry-Adams, you are a testimony of faith to me. Your encouragement has been tremendous. I thank you so much for your incessant prayers on my behalf, your heartening words, and your listening ears. Only in eternity we will both understand the import of your ministry to me, especially during this time.

Finally, to the all-wise God, who sees and knows all, I give You the greatest thanks my heart can muster. Truly, in You I live and move and have my being. Amen.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Active queue management (AQM) techniques for congestion control in Internet Protocol (IP) networks have been designed using both heuristic and analytical methods. For the latter, classical and modern control theory as well as deterministic optimization have been employed. These have resulted in stable and robust designs. At their core, however, are linearized, deterministic models of the Transmission Control Protocol (TCP). So far, there has been found no AQM scheme designed in the realm of stochastic optimization. Of the many options available in this arena, the gradient-based stochastic approximation method using Infintesimal Perturbation Analysis (IPA) gradient estimators is very promising. IPA requires no knowledge of the underlying probability distributions governing the system. It typically needs no knowledge of the traffic and service processes themselves. Additionally, IPA gleans all the data it needs for gradient estimation during a single run of the system (which can be 'live' or simulated) using fairly simple counting processes. Therefore, it can be considered an *online* estimator, which is quite appropriate for dynamic optimization.

More recent effort in IPA gradient estimation has been within the Stochastic Fluid Model (SFM) framework, but to date there have been no derivations for a tandem network of queues with loss-feedback. For any realistic AQM scheme, loss-feedback is a vital component, so it is essential that such derivations be performed. Additionally, this IPA-SFM for loss-feedback can readily lead to an AQM scheme that complements equation-based congestion control source algorithms for multimedia flows such as the TCP-Friendly Rate-Control (TFRC) algorithm. The research outlined in this thesis provides the theoretical basis and foundational layer for the development of IPA-based AQM schemes including those with loss feedback.

To set the context of the research performed, congestion control, the role of AQM in general, and existing AQM designs will first be briefly discussed. Stochastic optimization will then

be introduced. Next, the IPA methodology, with its benefits and disadvantages, will be presented together with previous research in the SFM framework. A very brief discourse on TFRC will ensue. This will be followed by a discussion of the research which includes the derivations of the algorithms for computing the IPA gradient estimators for the loss volume and queue workload for the following cases within the SFM framework:

- A single-stage queue with instantaneous, additive loss-feedback,
- A single-stage queue with instantaneous, additive loss-feedback and an unresponsive competing flow,
- A single-stage queue with delayed, additive loss-feedback, and
- A multi-stage tandem network of m queues with instantaneous, additive loss-feedback,

Sensitivity analyses and optimizations were performed with control parameter, θ, being the buffer-limits of the queue(s), as well as the loss-feedback constant.

CHAPTER I

INTRODUCTION

1.1 Congestion control in the Internet

Congestion-control research has been going on for three decades now. Because of the connectionless architecture of IP networks, and hence the infeasibility of preallocating resources end-to-end for each flow, congestion control in the Internet is a very complex problem [3]. Also contributing to this complexity are the ever-increasing size and heterogeneity of these networks. Congestion leads to high latency in data transfers, wasted resources resulting from packet losses, and the possibility of congestion collapse [3]. The goal of congestion control is to maintain stability, maximize throughput, and minimize delay and packet loss in the network [40]. To this end, the challenge of congestion control has been tackled from two fronts: the network's edge and its interior. On the network's edge there are end-system algorithms such as the Transmission Control Protocol (TCP). In the network's interior there are the various forms of Active Queue Management (AQM).

End-system algorithms for congestion control are typically implemented at the transport layer. The dominant transport layer protocol in the Internet is TCP. For congestion control, TCP end-points use dynamic windows to control the rates at which they inject data into the network [96]. TCP considers packet loss as an indicator of congestion and adjusts the sending rate when such congestion occurs. TCP has contributed to the robustness of the Internet [88]. However, there are limitations [3, 24, 46]. For the most part, end-point congestion control seeks to cure the network after congestion has occurred [3]. In other words, it is reactive [75]. There is also a delay between the packet-drop event at the router and the detection of this loss at the end-system. During this time lag, TCP will still continue to send at the high transmission rate, leading to an even higher number of packet losses [22, 70]. Additionally, TCP contributes to the bursty nature of Internet traffic. So, to absorb this burstiness and maintain high link utilizations and low packet

loss, the network must employ large buffers at its routers. But large buffers incur high queueing delays, especially when congestion occurs. In addition, multimedia and peer-to-peer applications are on the rise [59], and these demand QoS guarantees (e.g., low delay and low delay variance), which, on its own, end-point congestion control cannot fully provide. See also [89, 88].

1.2 The role of Active Queue Management

In general, queue management is the process by which a router decides which packets to drop when an output port becomes congested [7]. The traditional first-in-first-out (FIFO) droptail (simply referred to as "droptail" from now on) was initially the only queue management scheme in the Internet [75]. It is simple and easy to implement in routers [75]; however, it exacerbates the limitations of end-point congestion control schemes such as TCP.

Droptail queues cause 'lock-out' [59, 75], by which a small number of flows can grab a disproportionate share of the link bandwidth, blocking other competing flows. Also, when a droptail queue becomes full, it drops all further incoming packets, giving no advance warning of congestion to the end-systems. The packet drops can also be bursty, leading to network instability [19]. There is also the problem of global synchronization of TCP flows as a result of the high correlation among packet drops. These TCP flows might all simultaneously reduce their congestion windows and hence their sending rates [57]. This results in queue oscillations and contributes to high queuing-delay variance (also known as jitter). This jitter also interferes with the TCP self-clocking mechanism as the spacing between ACKs will vary more, causing TCP to be even more bursty [1]. Droptail queues also penalize flows with longer round-trip times [58]. Moreover, droptail is not suitable for real-time applications, which are delay sensitive.

On the other hand, AQM methods complement the end-point congestion control with the aim of preventing congestion [3]. AQM, unlike droptail queue management, is a proactive congestion-control scheme by which the network informs end-systems when there is incipient congestion [75]. It can do so explicitly in the form of Explicit Congestion Notification

(ECN) markings or implicitly by packet drops [59]. As the congestion level increases, the AQM scheme intensifies its feedback to the end-points (e.g., TCP) by marking or dropping more packets. The TCP end-points, in response to the congestion notification, reduce their transmission rates so as to prevent queue overflow and the consequent losses [96, 75]. Because the AQM algorithm acts within a router, the place where the congestion is actually taking place, it will obtain more accurate and timely congestion information than the traffic end-systems [75].

1.3 Existing AQM techniques and their design principles

A number of AQM techniques exist. The earliest of these is Random Early Detection (RED) [26]. RED has been rigorously analyzed and a number of its drawbacks have been cited. Numerous improvements have been proposed in the form of new RED variants (e.g., Gentle RED, Adaptive RED [24], Balanced RED [2], Flow-based RED [58], Stabilized RED [65], Dynamic RED [96]). Completely new schemes have also been proposed (e.g., BLUE [22], Stochastic Fair BLUE [21], GREEN [23]). The design of these schemes has been primarily heuristic, with parameter-tuning being one of the main drawbacks. As a result, there has also been a growing trend in the research community to design AQM techniques through analytical means so as to improve their stability and robustness. One main thrust in this direction has been to employ the tools of classical and modern control theory. To this end, there have been AQM schemes such as the Proportional Integral (PI) controller [38], Proportional-Integral-Derivative (PID) controller [91], Predictive PID controller [97], and Proportional Derivative (PD) RED [79, 44], just to name a few. At the heart of these control-theoretic schemes is a linearized fluid model of the TCP-Reno additive-increase-multiplicative-decrease (AIMD) behavior that attempts to characterize the dynamics of the traffic entering the queue controlled by the AQM. Besides the fact that these control-theoretic AQM schemes are TCP-centric, there are limitations inherent to the TCP modeling approach itself.

The first limitation is that the TCP model may not capture all of TCP behavior. For bulk data transfers, the slow-start phase is insignificant compared to the congestion-avoidance

phase so that modeling the congestion-avoidance phase alone will suffice. However, applying this model to Web traffic, for which the slow-start phase is indeed significant, would be even further from ideal. The second limitation arises from linearizing the non-linear TCP system around an operating point and assuming that the system is time-invariant. This results in inaccuracy if the operating point shifted dramatically during system operation, and there would be statistical errors by assuming stationarity.

Optimization theory has also been used to design AQM schemes. In particular, there is the deterministic optimization approach in which the AQM scheme, in conjunction with the end-system algorithm, explicitly attempts to drive the network to a globally optimal operating point [60]. Early contributors to this line of AQM research have been Frank Kelly (e.g., [42]) and Steven Low (e.g., [60]), among others. The optimization problem is formulated as the maximization of the sum of all the source utilities in the network subject to the constraint that the aggregate source rate at each link not exceed its capacity. It is assumed that the utility function of each source is increasing, strictly concave, and twice continuously differentiable with respect to source rate. It is also assumed that these utilities are additive. Based on these assumptions there will be a unique optimal solution for this primal problem. There are, however, two issues with this approach. First, the utility functions may not be known by the network and they may differ among the sources [42]. Second, solving the primal problem directly will require coordination among potentially all the sources (a complex and impractical situation) [60]. This is because the source rates are coupled by the constraint, although they are separable in the objective function. The more appropriate line of attack would be to maximize the Lagrangian, i.e., the sum of the maximum benefit each source enjoys (which is the difference between its utility and the bandwidth cost it incurs) and the network benefit (i.e., the price of bandwidth across all links). The Lagrange multiplier is the price per unit bandwidth. Kelly then used the "penalty-function" approach to solve the optimization problem by which the link price is assumed to be some pre-defined function of the aggregate flow rate through the link. He also assumed proportional fairness among the sources, i.e., a logarithmic utility function[1],

[1]To determine the utility functions inherent to the various flavours of TCP, deterministic flow models are

but later generalized the solution. On the other hand, Low solved the dual problem by finding the price that minimizes the Lagrangian. The algorithm he proposed used a gradient projection method to find the price. Low's approach obviates the need to know the utilities of the sources. The conceptual difference between Low's approach and Kelly's approach is that in the case of Kelly, the source decides how much it is willing to pay over each unit of time, and the network allocates its rate accordingly, whereas in Low's case, the source tells the network the rate it wants, and the network charges a price accordingly [60]. Based on this general algorithm by Low, the AQM, Random Early Marking (REM), was developed [4]. Kunniyur and Srikant developed the Adaptive Virtual Queue (AVQ) based on Kelly's work. See [49] and [48]. It should be noted that the deterministic optimization approach implicitly assumes no packet losses.

1.4 Stochastic optimization for AQM design

Another approach to optimization is stochastic optimization. To date, no AQM scheme built on the foundation of stochastic optimization, has been found. It may be worthwhile to explore those options available in this theoretical realm that will be most amenable to AQM design for communication networks.

In his survey Glynn presents a fundamental hierarchical framework for stochastic optimization [32]. The base classification is finite-dimensional stochastic optimization versus infinite-dimensional stochastic optimization. According to [32], for infinite-dimensional stochastic optimization one is trying to determine the optimal (usually time-varying) control policy among a permissible set of control policies that minimizes the system cost. On the other hand, finite-dimensional stochastic optimization attempts to find the optimal control parameter vector, $\hat{\theta}^* \in \Re^d$, $d \geq 1$ that belongs to the probability space $(\Omega, \mathcal{F}, P_\theta)$ and that minimizes the average system cost, $E[J(\theta)] = \int_\Omega J(\theta, \omega) dP_\theta(\omega)$, which may or may not be subject to constraints (and these in turn may also be functions of random vectors). A further classification can be made of finite-dimensional stochastic optimization, i.e., continuous parameter stochastic optimization versus discrete parameter optimization [32],

---

often used.

and these employ different techniques. In [81], the author classifies continuous parameter stochastic optimization even further into gradient-based and non-gradient-based techniques. Gradient-based approaches usually employ the general recursive Robbins-Monro stochastic approximation (SA) procedure to find the root of the gradient:

$$\hat{\theta}_{n+1} = \hat{\theta}_n - a_n \hat{g}(\hat{\theta}_n) \tag{1}$$

where $\hat{g}(\hat{\theta}_n)$ is the noisy estimate of the gradient at the iterate $\hat{\theta}_n$ and a_n is the step size (or gain). Conditions that a_n must satisfy include $a_n > 0$, $\sum_{n=1}^{\infty} a_n^2 < \infty$ and $\sum_{n=1}^{\infty} a_n = \infty$. One sequence of gains that satisfies these conditions is $a_n = a_0 n^{-1}$.

There are some optimization problems that can only be solved via computer simulation, and these are referred to as simulation optimization problems [5]. In these problems, the objective function and/or constraints cannot be expressed in a purely analytical form so that the exact expression for the gradient cannot be obtained [5]. Additionally, the objective function and constraints are stochastic in nature. However, efficiency is key since it is more expensive (in terms of time and computing resources) to run simulations than to solve analytical functions [5]. The various simulation optimization techniques that employ SA differ in the way the gradient itself is estimated, making some more or less reliable and efficient than others [5]. Three classical techniques for gradient estimation are Finite Differences (FD), Likelihood Ratio or Score Function (LR/SF), and Infinitesimal Perturbation Analysis (IPA). FD is the most elementary of the three, requiring no knowledge of the underlying system dynamics, whereas LR/SF and IPA exploit the structure of the system. At this juncture, the research question is whether or not a gradient-based stochastic optimization AQM technique can be derived for communication networks. Of the three approaches, IPA seems to be the most promising for reasons that are discussed in the following section.

1.5 Infinitesimal Perturbation Analysis

Following L'Ecuyer's work in [54, 53] (see also [33]), let us consider a simulation model defined over a probability space $(\Omega, \mathcal{F}, P)$ and as before, let the sample cost (i.e., after each simulation run) be $J(\theta, \omega)$, which is assumed to be measurable at each θ. For IPA, the

probability measure (or law) P is assumed to be independent of θ so that θ is taken as a purely structural control parameter rather than a distributional parameter [76]. Typically, the sample path, ω, is a sequence of independent and uniformly distributed random variables $(\zeta_1, \zeta_2, \ldots, \zeta_m)$, i.e., $\zeta_i \sim U(0,1),\ i = 1, \ldots, m$, for random variate generation in the simulation. Therefore, the expected cost is

$$E[J(\theta)] = \int_{\Omega} J(\theta, \omega) dP(\omega) \tag{2}$$

And taking derivatives with respect to θ, one gets

$$\frac{dE[J(\theta)]}{d\theta} = \int_{\Omega} \frac{dJ(\theta, \omega)}{d\theta} dP(\omega) = E\left[\frac{dJ}{d\theta}\right] \tag{3}$$

provided that $J(\theta, \omega)$ is a continuous function of θ so that $\frac{dJ(\theta,\omega)}{d\theta}$ is an unbiased estimator of $\frac{dE[J(\theta)]}{d\theta}$, i.e., the derivative and expectation operations can be interchanged [53, 30]. So that for IPA, it is possible to directly extract the necessary estimator of the gradient of the expected cost from only a single sample path (i.e., simulation replication). This is the most important advantage of the IPA technique since it requires less computing effort (and results in higher efficiency) for the optimization process [5]. However, this interchange, under a number of practical applications, cannot occur.

In the context of discrete-event dynamic systems (DEDS), the performance function depends on the timing of events on the sample path. When there is a perturbation in the control parameter, the timing of events can be affected by "perturbation generation" or "perturbation propagation" [37, 10]. By perturbation generation, the timing of an event (e.g., service completion time) is directly shifted because of the change in the control parameter. This shift in time for such an event can cause the shift in time for future events, i.e., perturbation propagation. Now, it is assumed that an infinitesimal perturbation of the control parameter θ will only result in the perturbation (or sliding forward) of the times of events but not in their sequence [53], hence guaranteeing that the gradient estimator is unbiased (i.e., the interchange of derivation and expectation can occur). However, there are examples of systems where even an infinitesimal change in θ will cause an inevitable (drastic) reordering of events and hence discontinuities in the performance function. These

include multiclass queueing networks where θ is the routing probability [53] and queueing networks with buffer limits.

Besides the interchange problem (with its restrictions on the applicability of IPA), another disadvantage of IPA is that, typically, IPA simulation models must be rebuilt from scratch for any modification [5]. Hence, reuse and modularization are not applicable. Also this re-work requires strong structural knowledge of the system model.

Although IPA may have been developed independently of LR/SF as a gradient estimation technique, L'Ecuyer in [52, 53] showed that IPA can be viewed as a special case of the LR/SF method. For IPA, it was assumed that the sample space with its probability law was independent of the control parameter θ. For LR, however, the more general case is considered, i.e., the probability law governing the sample space is dependent on θ.

In summary, IPA directly gleans all the data it needs to estimate the gradient of the performance function from a single arbitrary sample path (or simulation run); hence, there is no need for multiple replications at each value of the control parameter θ. Additionally, IPA requires no knowledge of the underlying probability distributions governing the system and typically needs no knowledge of the traffic and service processes themselves (i.e., rates etc.). In other words, it is non-parametric. Therefore, while passively observing the system as it runs, IPA can simultaneously compute the gradient estimate using fairly simple counting processes, leading to an efficient "online" estimator.

Initial work on IPA gradient estimation was in the context of DEDS. Much theoretical work had been developed using the traditional queueing paradigm, for which the enqueue and dequeue events for each "customer" must be accounted for in the derivation and implementation of the gradient estimator. The $M/M/1$, $M/D/1$, $M/U/1$, $M/G/1$, $G/G/1$, $M/M/m$, $G/D/m$ single-stage systems have been examined. A sample of such works include [18, 34, 31, 28, 8, 67, 55, 36, 63, 62, 9, 29, 17]. Typical performance measures that were considered include mean system time and mean waiting time. The control parameter is typically some parameter of the service-time and inter-arrival-time distributions (e.g., the mean).

More recent work on IPA gradient estimation has shifted to the fluid flow modelling arena,

which is at a higher level of abstraction. Instead of using the detailed queueing models (for which each entity can be individually identified), the time scale of the process is increased so that entities are aggregated into continuous flows that are characterised by random rates. As a result, only major events are examined, e.g., jumps in source rates, the queue becomes full, becomes empty, ceases to be full, ceases to be empty, instead of every "customer" arrival and departure from the system [47]. According to [61], the individual entities in the flow can be considered as "molecules"in a fluid and, although their identity is lost, the overall behavior of the system remains the same. If these events, particularly the jumps in source rates, occur at a much slower rate than "customer" arrivals, then there would be fewer events that need to be examined in the continuous flow model (later called stochastic fluid model (SFM)). Computational efficiency (and hence simulation speed) increases. This advantage is even more significant when dealing with high-speed communication networks for which millions of packets may flow through a router per second. (See also [80].) Another advantage of using the continous flow model approach is that all parameters are continuous so that IPA can be more naturally applied than in its discrete-event counterpart [61, 80]. In [61], the author claims that, based on preliminary work in the continous flow paradigm, the IPA gradient estimators tend to be unbiased and nonparametric, which is not usually the case for those IPA gradient estimators derived in the discrete-event arena.

It may be fitting at this point to distinguish between analytical SFM modeling for system analysis and fluid simulation. We shall consider specifically the IPA context. Originally, IPA gradient estimators were developed using discrete models and used in discrete-event simulators (which processed individual entities and their associated events one at a time). As mentioned in the previous paragraph, IPA gradient estimators can be developed using continuous flow models. These in turn can be used in a continuous-flow (fluid) simulator. Alternatively, the same IPA gradient estimators derived within the continuous-flow setting can be implemented in a discrete-event simulator. Hence, the issue of building fluid simulators is separate from building analytical fluid models, though they are both based on the same concepts. Our focus will be on the latter regime.

1.6 IPA with Stochastic Fluid Models

With regard to theoretical IPA work using continuous flow models, [13] derived unbiased, non-parametric IPA gradient estimators for the expected loss rate and average queue length for the single-stage case with a simple threshold-based admission control, i.e., the control parameter is the buffer size. In [85] the single-stage case was also examined for the same performance functions (loss rate, cumulative workload) not only with respect to buffer size, but also service rate and inflow rate. In [14], the authors examined the single-stage in which two types of traffic are admitted: uncontrolled and threshold-based buffer controlled. A number of works have been generated for nodes in tandem. In [77], the authors derived unbiased, non-parametric IPA gradient estimators for the loss volume and buffer workload with respect to the buffer limit of the first node in a tandem of single-class nodes. (See also [78].) The authors in [87] also looked at nodes in tandem but examined the loss volume and buffer workload not only with respect to the buffer size of the first node but also of the upstream node. (See also [84].) The authors in [64] examined tandem networks of two-class stochastic fluid models. [69], although cast in the field of communications is another theoretical contribution. It examined the case when multiple buffers are served by a single server using non-idling scheduling polices, i.e., it addressed a resource allocation problem. In the references previous to this, each buffer had its own server. In [69], the performance metrics of interest were the loss volume and the queue workload, and the control parameters were the buffer size as well as the bandwidth share for each buffer. The resulting IPA gradient estimators were found to be unbiased and non-parametric.

So far in this review of theoretical IPA work using stochastic fluid models, no feedback has been discussed. The notion of feedback in this context was first introduced by [92, 94]. Here, the incoming rate to a single-stage network was decreased instantaneously and additively based on a function of the state of the buffer (i.e., its length). The control parameter was again buffer size and the performance measures were loss rate and average (queue) workload. The derived gradient estimators were non-parametric and unbiased. The authors in [86] then considered the IPA gradient estimator for loss volume in the case of (delayed) retransmissions for the single-stage network and for which the control parameter was again

the buffer size. Here, some notion of delay in the feedback path was examined. In [95, 93], the authors dealt with multiplicative feedback on a single node SFM. In [95] the input rate was decreased multiplicatively by a factor c ($0 < c \leq 1$) if the queue length exceeded a certain threshold ϕ. They derived the IPA gradient estimators for the loss rate and average queue workload performance metrics with respect to the feedback gain parameter c instead of the usual buffer size. The IPA estimators were found to be unbiased. The authors in [93] extended the work of [95] by deriving the IPA gradient estimators for loss rate and throughput metrics with respect to the buffer threshold ϕ at which the multiplicative feedback will be invoked. They then used the IPA gradient estimators with respect to c and ϕ in a joint two-dimensional optimization procedure using SA to minimize a cost function of the weighted sum of loss rate and throughput. The author of [68] also derived IPA gradient estimators for average buffer content and throughput for the single-stage case with feedback control. However, the source model differed from that in [95, 93, 92] and [94] in that it was neither additive nor multiplicative but consisted of a hybrid automaton of two states, $\alpha_1(t)$ and $\alpha_2(t)$, where $\alpha_1(t)$ and $\alpha_2(t)$ are time-varying random processes independent of the control parameter - the buffer size. The source first transmits at $\alpha_1(t)$ and when the buffer becomes full, the source will then transition to the second state and begin transmitting at the next state $\alpha_2(t)$ some random delay τ seconds later, where τ is a linear function of the buffer size. Some random ζ seconds later, the buffer sees the new source rate $\alpha_2(t)$ and the cycle repeats. Now, ζ represents the propagation delay from source to buffer and is assumed to be independent of the control parameter, the buffer size. In addition to this feedback-controlled source there was also an uncontrolled traffic source with input rate $u(t)$. The author [68] claimed that one of its contribution to the feedback realm of IPA-SFM work is that it explicitly incorporates delay in the feedback to the source, which is a realistic inclusion, especially with regard to communication networks.

By using the buffer size as the control parameter that is updated at each step of the optimization process so as to minimize a weighted function of loss and delay within the network, there is the semblance of an AQM technique. But a vital component of any realistic AQM technique is loss-feedback to the sources. There has been no published theoretical

IPA-SFM work on loss-feedback for a tandem network of queues, or for the single-stage case, that can serve as the underpinning of such a gradient-based stochastic optimization AQM scheme.

In particular, this IPA-SFM for loss-feedback can readily lead to a gradient-based stochastic-optimization AQM scheme that complements equation-based congestion-control source algorithms for multimedia flows such as the TCP-Friendly Rate-Control (TFRC) algorithm. The demand for, the emergence of, and the traffic associated with multimedia applications in the Internet have been rapidly increasing [72, 11, 15, 82]. While, for the most part, real-time applications can tolerate low levels of loss due to inherent redundancy and through the use of powerful forward-error correcting (FEC) codes, they are highly delay-sensitive [72, 83]. They require low and predictable bounds in delay and delay variance [15]. Therefore, for multimedia, low and predictable delay and delay variation are much more important than reliability. TCP was designed primarily for bulk-data transfers, and its focus is reliability as opposed to delay. Besides this, TCP couples its congestion-control mechanism with its reliability-control mechanism. Furthermore, the congestion control itself causes too high a rate variation for multimedia traffic. All this makes TCP unsuitable for multimedia transport [90, 56, 25, 83, 15, 39]. As a result, many applications have resorted to the User Datagram Protocol (UDP) as their transport-layer protocol, but with UDP there is no congestion control – a threat to network stability. Consequently, a new class of congestion control mechanisms has emerged for multimedia that is "TCP-friendly", an example of which is equation-based TFRC algorithm.

1.7 TFRC

TFRC is specified in RFC 3448 [35]. It was first proposed in [27]. At the heart of TFRC is the control equation:

$$T = \frac{s}{RTT\sqrt{\frac{2bp}{3}} + t_{RTO}\left(3\sqrt{\frac{3bp}{8}}\right)p(1+32p^2)} \tag{4}$$

where T is the long-term average throughput, s the maximum segment size, RTT the estimated round-trip time, t_{RTO} the estimated retransmission timeout, b the number of

packets acknowledged by a single TCP acknowledgment (taken to be one (1)), and p the loss-event probability. This control equation was derived by Padhye et al. in [66].

At the receiver, TFRC calculates the loss-event probability p and sends this information in an acknowledgment packet. The sender, upon receipt of an acknowledgment packet, performs a number of tasks: it retrieves the calculated p; it determines the round-trip time sample,R_{sample}, by way of time-stamps stored in the acknowledgment packet and calculates the estimated round-trip time RTT; it estimates retransmission timeout, t_{RTO}; it calculates the maximum theoretical sending rate based on Eq. (4) which serves as the upper-bound on the sending rate; if the actual sending rate is less than this sending rate, the sender increases its sending rate; however, if the actual sending rate is greater, the sender decreases it.

The TCP control equation models the steady-state send rate of a bulk transfer TCP-Reno flow (e.g., FTP transfer). The model attempts to capture both the TCP fast retransmit mechanism and time-out mechanism; therefore, it spans a wide range of loss rates. The model focuses on the congestion avoidance mechanism of TCP. It views TCP congestion window phase in terms of *rounds*. A round is delimited by ACK receptions, and its duration is equal to an RTT. It is assumed that the RTT and loss rate are independent of estimated sending rate, which only holds when there is a high level of statistical multiplexing at a bottleneck link. It is also assumed that packet losses among rounds are independent; however, packet losses within rounds are correlated. In the latter case, if a packet is lost, all remaining packets transmitted until the end of that round are also lost.

1.8 Contribution

Within the SFM framework, algorithms for computing the IPA gradient estimators for the loss volume and queue workload for a multi-stage tandem network of m queues with instantaneous, additive loss-feedback, have been derived. Sensitivity analysis and optimizations were performed with the control parameter, θ, being the buffer-limit at any queue along the tandem, as well as the loss-feedback constant. As preliminary steps, the IPA gradient estimators for the single-stage case having instantaneous, additive loss-feedback and delayed,

additive loss-feedback were derived. The single-stage case with a competing unresponsive flow was also examined.

1.8.1 Significance

This research provides the theoretical basis and foundational layer for the development of AQM schemes to be designed using the formal IPA gradient-based stochastic approximation method. These schemes will exploit the unique benefits of this form of optimization that no other approach can provide in the context of communication systems. And, as mentioned earlier, it can lead to AQM schemes that complement equation-based congestion-control source algorithms for multimedia flows.

1.8.2 Organization

From here on, the thesis is divided into three parts. In part 1 (Chapters 2, 3, and 4), the IPA for the single-stage case with loss-feedback is covered, in part 2 (Chapter 5), that for the multi-stage tandem with loss-feedback is presented, and in part 3, (Chapters 6 and 7), practical implementation considerations are discussed.

In Chapter 2, the IPA for a single-stage queue with instantaneous, additive loss-feedback is presented. In Chapter 3, is derived the IPA for a single-stage queue with not only instantaneous, additive loss-feedback but with an unresponsive competing flow as well. The IPA for the single-stage queue with delayed, additive loss-feedback is derived in Chapter 4. In Chapter 5, the IPA for a multi-stage tandem with instantaneous loss-feedback is dealt with. In particular, we derive the IPA gradient estimators for loss volume and queue workload with the control parameters either being the buffer capacity at any of the queues in the tandem or the common loss-feedback constant of the tandem network.

In Chapters 2 to 5, sensitivity analyses of the IPA gradient estimators and the optimizations were carried out in a fluid-flow setting using Matlab. However, to determine how well these estimators will perform in the more realistic packetized domain, we perform the sensitivity analyses and optimizations using the Georgia Tech Network Simulator (GTNetS). These results are presented and discussed in Chapter 6.

Although this thesis seeks only to provide the theoretical underpinnings for an AQM scheme

that is based on IPA optimization, a discussion of what one must consider when actually implementing such an AQM can be found in Chapter 7.

CHAPTER II

SINGLE-STAGE FLUID QUEUE WITH LOSS FEEDBACK

Initial IPA sensitivity analysis and optimization with additive loss feedback was performed on the single-stage fluid queue. It consists of a single server with a time-varying stochastic service rate $\beta(t)$ that services a buffer of finite capacity, b. The inflow rate is that of a single IPA-controlled source, and is denoted as the stochastic process $\alpha(t)$. The inflow rate is the original intended source rate, $\sigma(t)$ reduced additively by the product of the loss rate, $\gamma(t)$, at the queue and the feedback constant c. Additionally, there is the buffer occupancy denoted by $x(t)$ and the outflow rate denoted by $\delta(t)$. The sensitivity analysis and optimization are performed with respect to a control parameter, θ, which will be elaborated upon in the following subsections. Since $x(t)$, $\gamma(t)$ and $\delta(t)$ can be dependent on θ, we designate them as $x(\theta;t)$, $\gamma(\theta;t)$ and $\delta(\theta;t)$ respectively. Figure 1 depicts these different parameters and processes. The IPA time-horizon is taken as some constant T such that $T > 0$ and $t \in [0, T]$.

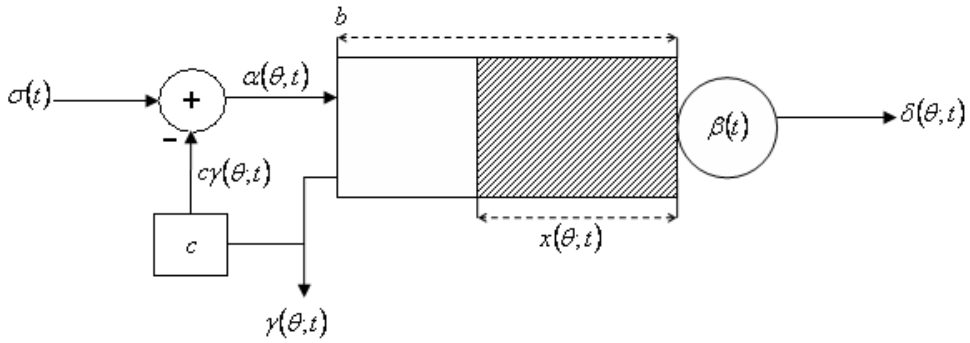


Figure 1: Stochastic fluid model with instantaneous additive-loss feedback for the single stage case

Define:

$$A(\theta;t) = \alpha(\theta;t) - \beta(t) \tag{5}$$

then:

$$\frac{dx}{dt} = \begin{cases} 0 & \text{if } x(\theta;t) = 0 \text{ and } A(\theta;t) < 0 \\ 0 & \text{if } x(\theta;t) = b \text{ and } A(\theta;t) > 0 \\ A(\theta;t) & \text{otherwise} \end{cases} \tag{6}$$

also:

$$\gamma(\theta;t) = \begin{cases} A(\theta;t) & \text{if } x(\theta;t) = b \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

$$\lambda(t) = \sigma(t) - \beta(t) \tag{8}$$

Now:

$$\begin{aligned} \alpha(\theta;t) &= \sigma(t) - c\gamma(\theta;t) \\ \Rightarrow A(\theta;t) &= \lambda(t) - c\gamma(\theta;t) = \begin{cases} \frac{\lambda(t)}{1+c} & \text{if } x(\theta;t) = b \\ \lambda(t) & \text{otherwise} \end{cases} \end{aligned} \tag{9}$$

Using the relation in Eq. (9), Eq. (6) and Eq. (7) can be rewritten as:

$$\frac{dx}{dt} = \begin{cases} 0 & \text{if } x(\theta;t) = 0 \text{ and } \lambda(t) < 0 \\ 0 & \text{if } x(\theta;t) = b \text{ and } \frac{\lambda(t)}{1+c} > 0 \\ \lambda(t) & \text{otherwise} \end{cases} \tag{10}$$

$$\gamma(\theta;t) = \begin{cases} \frac{\lambda(t)}{1+c} & \text{if } x(\theta;t) = b \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

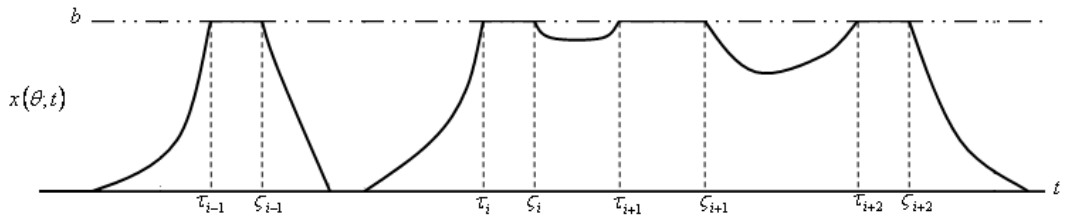


Figure 2: Trajectory of buffer-occupancy for the single-stage case

Consider the buffer-occupancy process in Figure 2. Let $\tau_i(\theta)$ be the time of the beginning of the i-th full period and let $\varsigma_i(\theta)$ be the time of the end of the i-th full period. Then the total loss volume, denoted as $L_T(\theta)$, is given by:

$$L_T(\theta) = \sum_{i=1}^{K} L_i(\theta) \tag{12}$$

where K is the number of full-periods in the interval $[0, T]$ and

$$\begin{aligned} L_i(\theta) &= \int_{\tau_i(\theta)}^{\varsigma_i(\theta)} \gamma(\theta; t)\, dt \\ &= \tfrac{1}{1+c} \int_{\tau_i(\theta)}^{\varsigma_i(\theta)} \lambda(t)\, dt \end{aligned} \tag{13}$$

The loss derivative is then given by

$$\frac{dL_T(\theta)}{d\theta} = \sum_{i=1}^{K} \frac{dL_i(\theta)}{d\theta} \tag{14}$$

At this juncture, we define two types of events: exogenous events and endogenous events. An exogenous event is a discontinuity in the defining processes $\sigma(t)$ and $\beta(t)$. On the other hand, an endogenous event is the beginning of an empty-period or full-period at the queue. Additionally, to ensure that the sample derviatives exist, the following assumptions are made:

(1) $\sigma(t)$ and $\beta(t)$ and piecewise continuously differentiable on the interval $[0, T]$. The number of points where they are discontinuous have finite first moments, and the number of points where their first derivatives change signs have finite first moments.

(2) No empty period or full period consists of a single point.

(3) Exogenous events cannot occur simultaneously with other exogenous events or with other endogenous events.

(4) During an open sub-interval of any boundary period it never occurs that $A_m(\theta; t) = 0$

2.1 Buffer capacity, b, as the control parameter, θ

$$\begin{aligned} \tfrac{dL_i(\theta)}{d\theta} &= \tfrac{1}{1+c} \tfrac{d}{d\theta} \left[\int_{\tau_i(\theta)}^{\varsigma_i(\theta)} \lambda(t)\, dt \right] \\ &= \tfrac{1}{1+c} \left[\int_{\tau_i(\theta)}^{\varsigma_i(\theta)} \tfrac{d\lambda(t)}{d\theta}\, dt + \lambda(\varsigma_i(\theta)^-) \tfrac{d\varsigma_i(\theta)}{d\theta} - \lambda(\tau_i(\theta)^+) \tfrac{d\tau_i(\theta)}{d\theta} \right] \end{aligned} \tag{15}$$

Since $\lambda(t) = \sigma(t) - \beta(t)$ and $\sigma(t)$ and $\beta(t)$ are independent of θ, then $\lambda(t)$ is independent of θ within the interval $(\tau_i(\theta), \varsigma_i(\theta))$ so that $\int_{\tau_i(\theta)}^{\varsigma_i(\theta)} \frac{d\lambda(t)}{d\theta}\, dt = 0$

Consider, $\varsigma_i(\theta)$, the time when the queue ceases to be full. This can happen in either of two ways. In the discontinuous case, there is a jump in $A(\theta;t)$ from $\frac{\lambda(\varsigma_i(\theta)^-)}{1+c}$ (a positive number) to $\lambda(\varsigma_i(\theta)^+)$ (a negative number) and this can only occur if there is a sign change in $\lambda(t)$ at $t=\varsigma_i(\theta)$ regardless of the value of $(1+c)$. This sign change will be due to a jump in the defining processes $\sigma(t)$ and $\beta(t)$ which is independent of θ. Therefore $\frac{d\varsigma_i(\theta)}{d\theta}=0$.
In the continuous case, $A(\theta;t)=\frac{\lambda(t)}{1+c}$ decreases continuously to zero at $t=\varsigma_i(\theta)$, which implies that $\lambda(t)$ decreases continuously to zero so that $\lambda(\varsigma_i(\theta))=0$ and then continues negatively. Therefore $\lambda(\varsigma_i(\theta)^-)\frac{d\varsigma_i(\theta)}{d\theta}=0$.
Eq. (15) then becomes:

$$\frac{dL_i(\theta)}{d\theta}=-\frac{1}{1+c}\left[\lambda(\tau_i(\theta)^+)\frac{d\tau_i(\theta)}{d\theta}\right] \tag{16}$$

Now, consider the non-boundary period (NBP) that precedes the full-period. That NBP may have itself been preceded by an empty period or a full-period. In the first case it will be designated as *EF* and the latter as *FF*.

$$\begin{aligned} x(\theta;t)|_{\varsigma_{i-1}(\theta)}^{\tau_i(\theta)} &= x(\theta;\tau_i(\theta))-x(\theta;\varsigma_{i-1}(\theta))=\begin{cases}\theta & \text{if } NBP \text{ is } EF\\ 0 & \text{if } NBP \text{ is } FF\end{cases}\\ &= \textstyle\int_{\varsigma_{i-1}(\theta)}^{\tau_i(\theta)}\frac{dx(\theta,t)}{dt}\,dt\\ &= \textstyle\int_{\varsigma_{i-1}(\theta)}^{\tau_i(\theta)}\lambda(t)\,dt \end{aligned} \tag{17}$$

Taking derivatives of Eq. (17) with respect to θ:

$$\lambda(\tau_i(\theta)^-)\frac{d\tau_i(\theta)}{d\theta}=\begin{cases}1 & \text{if } NBP \text{ is } EF\\ 0 & \text{if } NBP \text{ is } FF\end{cases} \tag{18}$$

Since $t=\tau_i(\theta)$ corresponds to an endogenous event, i.e. queue becomes full, and we have already assumed that an endogenous and exogenous events cannot simultaneously occur, then $\lambda(\tau_i(\theta)^-)=\lambda(\tau_i(\theta)^+)$ so that

$$\frac{dL_i(\theta)}{d\theta}=\begin{cases}-\frac{1}{1+c} & \text{if preceding } NBP \text{ is } EF\\ 0 & \text{if preceding } NBP \text{ is } FF\end{cases} \tag{19}$$

Therefore, if there are N_T *lossy busy-periods*[1] in the interval $[0, T]$, then

$$\frac{dL_T(\theta)}{d\theta} = -\frac{N_T}{1+c} \tag{20}$$

We see that one needs to simply keep a running count of the number of lossy busy-periods and scale by the factor $-\frac{1}{1+c}$.

Because the loss-feedback is felt only when the queue is in a full-period, the IPA derivative for the queue workload is the same as that of the single-stage case with no loss-feedback. For the latter, the derivation can be found in [85].

2.1.1 A note on unbiasedness

When the following two conditions are both met, the IPA derivative estimators will be unbiased [74]. Consider the IPA loss derivative[2]:

1. For every $\theta \in \Theta$ where Θ is a closed and bounded set, the sample derivative, $\frac{dL(\theta)}{d\theta}$ exists w.p.1.

2. W.p.1 the sample function, $L(\theta; T)$ is Lipschitz continuous in θ, and its Lipschitz constant has a finite first moment.

The first condition follows from the assumptions specified between Eq. (14) and Eq. (15). If the following is true then the second condition is satisfied.

$$\max\left\{\left|\frac{dL}{d\theta}(\theta)\right|\right\} \quad < \quad K \tag{21}$$

where $K > 0$ is a random variable with $E[K] < \infty$. In other words, the maximum magnitude of the sample derivative over all θ should always be less than some constant, K, which is the Lipschitz constant and the mean of which is always finite. This Eq. (21) can be proven simply by applying the mean-value theorem.

In all, what remains to be done, is to determine that K for each sample-derivative we derived using IPA, to prove unbiasedness. In particular, for this single-stage case with instantaneous

[1]Note that there is a difference between the full period and the lossy busy-period. A lossy busy-period comprise a number of non-boundary periods and full-periods, so that there may be more than one full-period in a single lossy busy-period

[2]This applies also to the queue-workload derivative

additive loss feedback and with buffer length being the control parameter of interest, we find that $\left|\frac{dL}{d\theta}(\theta)\right| < N_T$ where N_T is the number of lossy busy-periods in the interval $[0,T]$. N_T is a random variable. Within the interval $[0,T]$, the independent processes, $\sigma(t)$ and $\beta(t)$, have only a finite number of discontinuities, so that the number of sign changes in the net inflow-rate $A(\theta;t)$ is also finite. As a result, $N_T > 0, E[N_T] < \infty$. Therefore, there is a $K > 0$ such that, for all $\theta \in \Theta$, $N_T \leq K$ and $E[K] < \infty$ and the IPA loss derivative is unbiased. The unbiasedness of the queue-workload derivative for this case was established in [85].

2.2 Feedback constant, c, as the control parameter, θ

$$\begin{aligned}
\tfrac{dL_i(\theta)}{d\theta} &= \tfrac{d}{d\theta}\left[\int_{\tau_i(\theta)}^{\varsigma_i(\theta)} \tfrac{\lambda(t)}{1+\theta}\, dt\right] \\
&= \left[\int_{\tau_i(\theta)}^{\varsigma_i(\theta)} \tfrac{d}{d\theta}\left(\tfrac{\lambda(t)}{1+\theta}\right) dt + \left(\tfrac{\lambda(\varsigma_i(\theta)^-)}{1+\theta}\right)\tfrac{d\varsigma_i(\theta)}{d\theta} - \left(\tfrac{\lambda(\tau_i(\theta)^+)}{1+\theta}\right)\tfrac{d\tau_i(\theta)}{d\theta}\right] \qquad (22)\\
&= \int_{\tau_i(\theta)}^{\varsigma_i(\theta)} -\tfrac{\lambda(t)}{(1+\theta)^2}\, dt + \tfrac{1}{1+\theta}\left(\lambda(\varsigma_i(\theta)^-)\tfrac{d\varsigma_i(\theta)}{d\theta} - \lambda(\tau_i(\theta)^+)\tfrac{d\tau_i(\theta)}{d\theta}\right)
\end{aligned}$$

Consider the term $\lambda(\varsigma_i(\theta)^-)\frac{d\varsigma_i(\theta)}{d\theta}$ where $\varsigma_i(\theta)$ is the time the buffer ceases to be full. In the discontinuous case, the rate net-rate into the buffer (i.e. $A(\theta;t)$) must jump from a positive value of $\frac{\lambda(\varsigma_i(\theta)^-)}{1+\theta}$ to a negative value of $\lambda(\varsigma_i(\theta)^+)$. Given that c comes into effect only in the full period, the sign change would only come when there is a sign change in $\lambda(t)$, which will really be an exogeneous event. Therefore $\frac{d\varsigma_i(\theta)}{d\theta} = 0$.

For the continuous case, the same argument holds as when $b = \theta$, i.e. $\lambda(t)$ decreases continously to zero so that $\lambda(\varsigma_i(\theta)) = 0$ and then continues negatively. Therefore $\lambda(\varsigma_i(\theta)^-)\frac{d\varsigma_i(\theta)}{d\theta} = 0$.

As a result, Eq. (22) becomes:

$$\tfrac{dL_i(\theta)}{d\theta} = \int_{\tau_i(\theta)}^{\varsigma_i(\theta)} -\tfrac{\lambda(t)}{(1+\theta)^2}\, dt - \tfrac{1}{1+\theta}\left(\lambda(\tau_i(\theta)^+)\tfrac{d\tau_i(\theta)}{d\theta}\right) \qquad (23)$$

Again, consider the non-boundary period (*NBP*) that precedes the full-period. Equation (17) still holds, but taking its derivative with respect to θ will yield:

$$\lambda(\tau_i(\theta)^-)\frac{d\tau_i(\theta)}{d\theta} = 0 \qquad (24)$$

regardless of the type of *NBP*.

Due to continuity of $\lambda(t)$ at $t = \tau_i(\theta)$, $\lambda(\tau_i(\theta)^-) = \lambda(\tau_i(\theta)^+)$. Therefore

$$\frac{dL_i(\theta)}{d\theta} = \int_{\tau_i(\theta)}^{\varsigma_i(\theta)} -\frac{\lambda(t)}{(1+\theta)^2}\, dt \tag{25}$$

Now, substituting Eq. (13) into Eq. (25), we obtain a first-order differential equation with respect to θ, i.e.:

$$\frac{dL_i(\theta)}{d\theta} = -\frac{1}{1+\theta} L_i(\theta) \tag{26}$$

If all the full periods in the interval $[0, T]$ are considered so that we substitute Eq. (26) into Eq. (14), then Eq. (14) becomes

$$\frac{dL_T(\theta)}{d\theta} = -\frac{1}{1+\theta} L_T(\theta) \tag{27}$$

Solving Eq. (27):

$$\begin{aligned} \frac{dL_T}{d\theta} &= -\frac{1}{1+\theta} L_T \\ \Rightarrow \int \frac{1}{L_T} dL_T &= \int -\frac{1}{1+\theta} d\theta \\ \ln(L_T) &= -\ln(1+\theta) + K \\ \ln(L_T) &= \ln\left(\frac{A}{1+\theta}\right) \\ L_T &= \frac{A}{1+\theta} \end{aligned} \tag{28}$$

When $\theta = 0$ then $L_T(\theta) = L_T(0) = A$. Therefore

$$L_T(\theta) = \frac{L_T(0)}{1+\theta} \tag{29}$$

where $L_T(0)$ is the loss volume when there is no loss feed-back (i.e. the open-loop case). We see that the derivative $\frac{dL_T}{d\theta}$ is the total loss volume scaled by the factor $-\frac{1}{1+c}$ where $\theta \equiv c$.

Consider as a measure of error:

$$\epsilon = \frac{\frac{L_T(\theta+\Delta\theta) - L_T(\theta)}{\Delta\theta} - \frac{dL_T(\theta)}{d\theta}}{\left|\frac{dL_T(\theta)}{d\theta}\right|} \tag{30}$$

where the first term in the numerator is the first-order approximation to $\frac{dL_T(\theta)}{d\theta}$. Substituting Eq. (27) into Eq. (30) and simplifying:

$$
\begin{aligned}
\epsilon &= \frac{\frac{L_T(\theta+\Delta\theta)-L_T(\theta)}{\Delta\theta} + \frac{L_T(\theta)}{1+\theta}}{\left|-\frac{L_T(\theta)}{1+\theta}\right|} \\
&= \frac{\frac{L_T(0)}{(1+\theta+\Delta\theta)(\Delta\theta)} - \frac{L_T(0)}{(1+\theta)(\Delta\theta)} + \frac{L_T(0)}{(1+\theta)^2}}{\frac{L_T(0)}{(1+\theta)^2}} \\
&= \frac{\Delta\theta}{1+\theta+\Delta\theta} \\
&= \frac{1}{1+\frac{1+\theta}{\Delta\theta}}
\end{aligned}
\tag{31}
$$

The queue workload is independent of changes in c, the feedback constant, therefore $\frac{dQ_T(\theta)}{d\theta} = 0$ for all θ.

2.3 Simulations with buffer capacity, b, as the control variable θ

To verify the accuracy of the IPA gradient estimators derived for the single-stage fluid model with loss feedback, the test configuration as shown in Figure 3 was used.

The first queue calculates the IPA loss and queue-workload derivatives. There is no queueing at the second queue, Q_2, since $\beta_1(t) < \beta_2(t)$ for all $t \in [0, T]$ where $T = 10$ seconds. Nevertheless, the Q_2 is a simple first-in-first-out droptail queue. The service rate of the first queue, $\beta_1(t)$, is a random process, uniformly distributed between 6.75 Mbps and 11.25 Mbps. Its average service rate is 9 Mbps. The time interval between changes in the magnitude of $\beta_1(t)$ is a constant at 75 ms. The inflow rate, $\sigma(t)$ switches between either of two levels, 13.31 Mbps ("ON"-rate) and 3.32 Mbps ("OFF"-rate) every 100 ms or so. Its average is 8.32 Mbps. The feedback constant, c is 0.3333. The packet size is 554 bytes (512 bytes for the payload, and 42 bytes for the header).

2.3.1 IPA derivative error-analysis

The error-analysis was first carried out using Matlab and then using GTNetS (the Georgia Tech Network Simulator). For the latter, see Chapter 6. In Matlab, bit-level fluidization is almost realized. The limitation arises from the euler-step size (taken as $5\mu s$) in the numerical computation of the differential equations. GTNetS lowest level of granularity is the size of a packet, which is the case for real networks. Hence there is a high level of

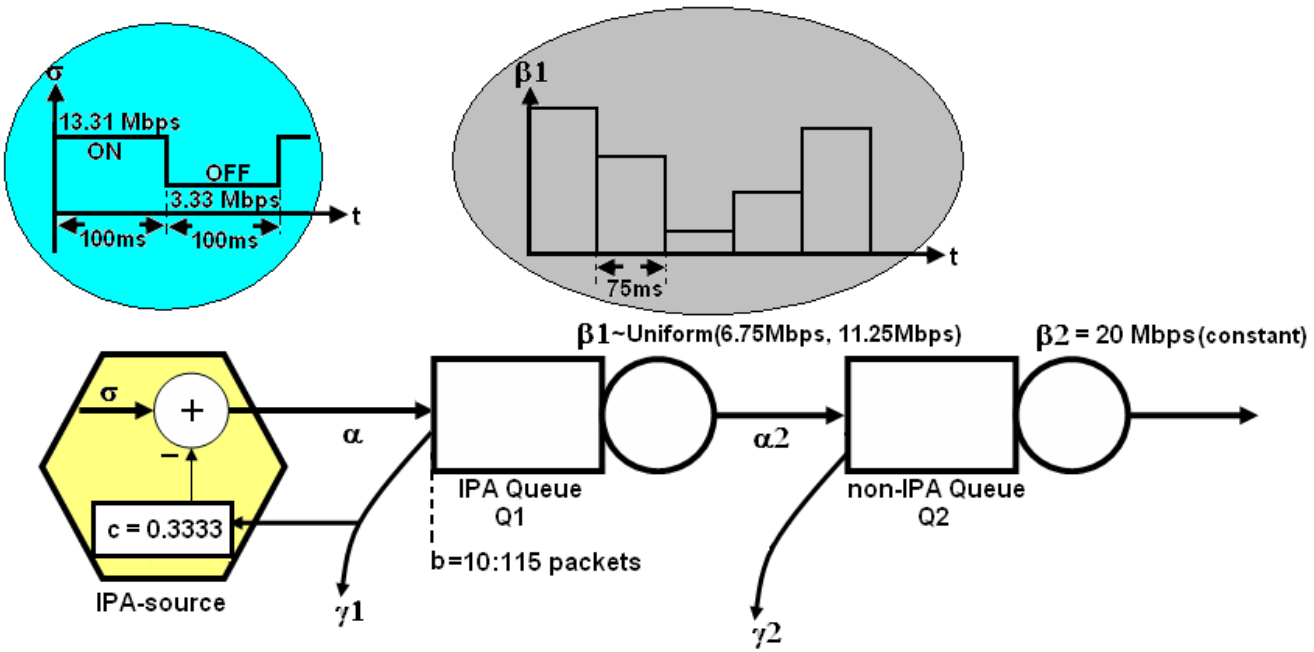


Figure 3: Test configuration for the single-stage case

discretization in the GTNetS implementation. Therefore, Matlab was used to verify the mathematical accuracy of the IPA derivatives, whereas GTNetS was used to see how well these IPA derivatives perform in a discrete setting. The latter will be discussed in Chapter 6. As the control variable, the buffer capacity of the Q_1 was incremented by 1 packet at a time for GTNetS (or $554 \times 8 = 4432$bits for Matlab) (i.e. $\Delta\theta = 1$ packet) from 10 packets to 150 packets. The same seed for the random-number generator was used for each value (i.e iterate) of the control variable. At each iterate, the error in the IPA loss-derivative ($\frac{dL_T}{d\theta}$)was calculated as in Eq. (30).

Besides the IPA-derivative error, the cost funtion was also calculated for each value of the control variable. The cost function was taken as $\frac{w_L}{T}L_T + \frac{w_Q}{T}Q_T$. Here the weights were chosen to be $w_L = 10$ and $w_Q = 70$.

The IPA fractional loss-derivative error using Matlab was plotted against θ and is shown in Figure 4a. The plot of IPA fractional queue-workload derivative error using Matlab versus θ is shown in Figure 4c. It can be seen that for increasing values of θ the fractional error increased, this was predominantly due to decreasing magnitudes of the IPA loss derivative (see the denominator of Eq. (30)) which amplified the same difference between the measured and the IPA derivative in the numerator of Eq. (30). To clarify this, the measured derivative (i.e. $\frac{L_T(\theta+\Delta\theta)-L_T(\theta)}{\Delta\theta}$ or $\frac{Q_T(\theta+\Delta\theta)-Q_T(\theta)}{\Delta\theta}$) and IPA derivatives were plotted on the same axes as in Figure 4b and Figure 4d. It can be seen that the actual derivatives do track well.

As another way of testing the accuracy of the IPA derivative estimators, we ran simulations for a long period (i.e., $T = 240$ seconds) for three sets of four adjacent values of θ (in packets), namely $\{29, 30, 31, 32\}$, $\{50, 51, 52, 53\}$ and $\{99, 100, 101, 102\}$, and computed the IPA derivative in each case. Using the first-order approximation of the Taylor series expansion as in Eq. (30) we calculated the error. This process was repeated for three different seed values for the random number generator. The error values are presented in Table 1.

From Table 1, it can be seen that in most cases, the magnitude of the percentage error in the loss derivative was much less than 0.01% with the exception for the group $b = \{99, 100, 101, 102\}$, where it varied between 1.10% and 2.7% which is still a low value. The magnitude of the percentage error in the queue-workload derivative was, for the most part,

less than 0.2%, except for $b = \{99, 100, 101, 102\}$ where it was as high as 1.7%.

2.3.2 IPA optimization

The optimization was carried out using Matlab. Two types of optimizations were performed. In the first case, the same seed of the random generator was used at each iteration of the optimization algorithm. In the second case, each iteration was independent of the next. For the first case, the optimization was carried out for 40 iterations, whereas for the second case it was carried out for 120 iterations. For both cases, the time horizon for each iteration was $T = 10$ seconds.

Using the same seed for the random generator as in the first case, the cost was plotted against the control variable θ. This is shown in Figure 5a. The minima is at $b = 90$ packets. Hence, if the derivatives were correct, then the optimization using the same seed (which makes it essentially deterministic) should converge to that value. Figure 5b shows this to be true. The results for the optimization in a fully random setting are shown in Figure 5c. There is convergence to the value slightly greater than $b = 90$ packets.

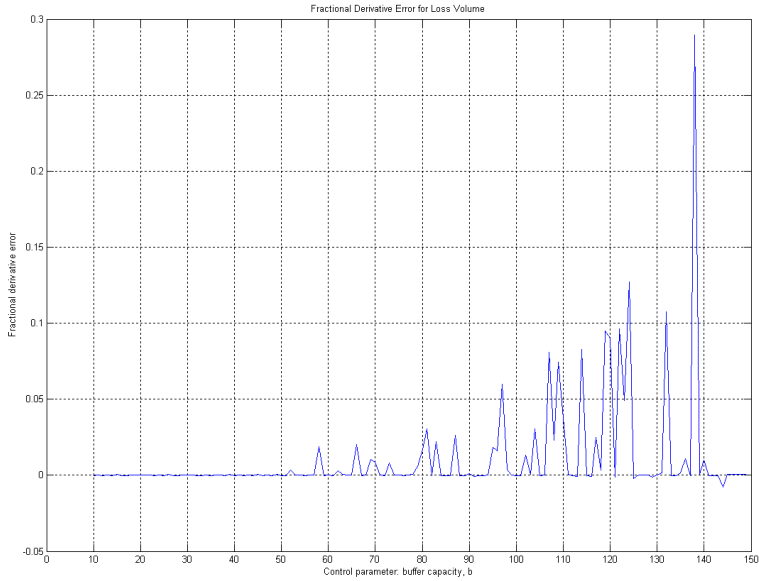
The control variable θ (i.e. the buffer-limit in Q_1) was adjusted after each run, i, and was used as the buffer-limit in the next run (i.e., $i + 1$). The algorithm for this adjustment was as follows:

$$\theta_{i+1} = \theta_i - a_i \frac{dJ}{d\theta}(\theta_i) + \Delta \tag{32}$$

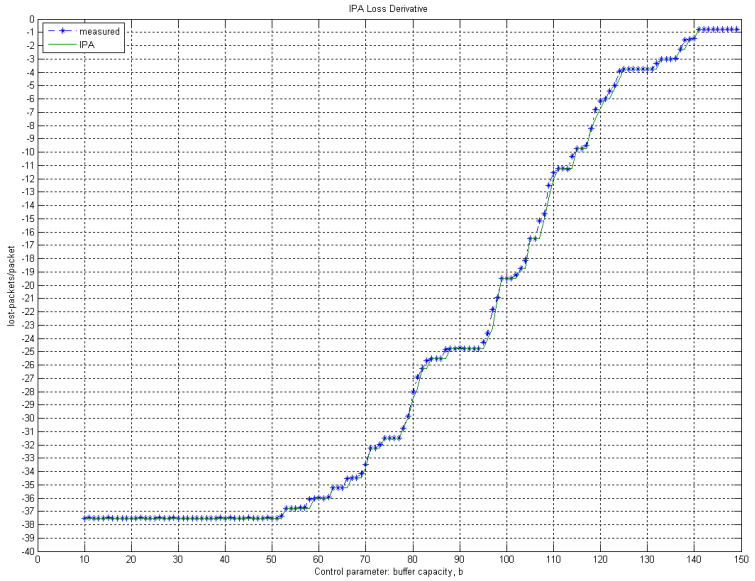
- if $|\lceil \theta_{i+1} \rceil - \theta_{i+1}| \leq |\lfloor \theta_{i+1} \rfloor - \theta_{i+1}|$
  - $\diamond$ $\Delta = \theta_{i+1} - \lceil \theta_{i+1} \rceil$
  - $\diamond$ $\theta_{i+1} = \lceil \theta_{i+1} \rceil$
- else
  - $\diamond$ $\Delta = \theta_{i+1} - \lfloor \theta_{i+1} \rfloor$
  - $\diamond$ $\theta_{i+1} = \lfloor \theta_{i+1} \rfloor$
- if $\theta_{i+1} < 2, \theta_{i+1} = 2$.
- if $\theta_{i+1} > 2000, \theta_{i+1} = 2000$

Table 1: IPA sensitivity analysis - the single-stage case $\theta \equiv b$

	seed=1000					
b (pkts)	$\frac{dL}{d\theta}$	L(bits)	$\epsilon_L(\%)$	$\frac{dQ}{d\theta}$	Q(bits-seconds)	$\epsilon_Q(\%)$
29	-900.002	272822987.1	0.00422	109.861815	14748992.87	-0.14876
30	-900.002	268834345.5	-0.00814	109.534925	15235176.09	-0.15017
31	-900.002	264845210.9	0.00617	109.205485	15719905.88	-0.15074
32	-900.002	260856646.9		108.876425	16203175.02	
50	-897.752	189071701.2	0.05397	103.268055	24660634.75	-0.14965
51	-896.252	185095010.7	0.14649	102.932695	25117633.86	-0.20917
52	-892.502	181128639.4	0.05120	102.4415	25572877.34	-0.14790
53	-891.002	177175094.6		102.114715	26026226.57	
99	-373.501	35738024.2	2.38485	64.550435	43400424.8	-1.52120
100	-357.751	34122145.84	1.52403	63.03138	43682160.37	-0.83207
101	-347.251	32560758.13	1.10837	62.34028	43959191.01	0.04333
102	-339.751	31038800.24		61.87638	44235602.84	
	seed=5586					
b (pkts)	$\frac{dL}{d\theta}$	L(bits)	$\epsilon_L(\%)$	$\frac{dQ}{d\theta}$	Q(bits-seconds)	$\epsilon_Q(\%)$
29	-900.002	269363563.4	0.01198	109.35815	14715957.37	-0.15638
30	-900.002	265375231.3	-0.00533	109.01513	15199874.77	-0.15873
31	-900.002	261386208.7	-0.00035	108.669275	15682262.91	-0.15897
32	-900.002	257397384.7		108.32364	16163119.49	
50	-897.002	185629128.7	0.01971	102.386935	24563053.24	-0.16922
51	-896.252	181654398.5	0.11993	102.02359	25016064.26	-0.19152
52	-894.752	177686972.3	0.13334	101.653045	25467366.83	-0.20005
53	-893.252	173726718		101.28434	25916991.83	
99	-383.251	34089501.12	2.25909	61.418735	42773237.7	-1.10570
100	-364.501	32429305.1	1.80773	60.179405	43042435.73	-0.90427
101	-351.751	30843040.35	2.72423	58.98256	43306739.02	-1.74851
102	-335.251	29326550.07		57.34072	43563578.93	
	seed=9736					
b (pkts)	$\frac{dL}{d\theta}$	L(bits)	$\epsilon_L(\%)$	$\frac{dQ}{d\theta}$	Q(bits-seconds)	$\epsilon_Q(\%)$
29	-900.002	272462598.1	-0.00498	109.9274	14754040	-0.15137
30	-900.002	268473589.3	0.00187	109.5947	15240501	-0.15069
31	-900.002	264484853.9	0.00042	109.2648	15725493	-0.14877
32	-900.002	260496060.7		108.9394	16209034	
50	-899.252	188703877.2	-0.00543	103.4254	24673044	-0.13649
51	-899.252	184718175	0.01504	103.1427	25130800	-0.14445
52	-897.752	180733288.3	0.10770	102.788	25587268	-0.17888
53	-894.002	176758735.6		102.3194	26042010	
99	-387.001	35225694.55	2.26351	64.82311	43340842	-0.48263
100	-369.751	33549329.8	1.63165	63.81867	43626752	-1.04375
101	-356.251	31937332.2	1.83673	62.75323	43906644	-0.32710
102	-342.001	30387428.39		61.92831	44183857	

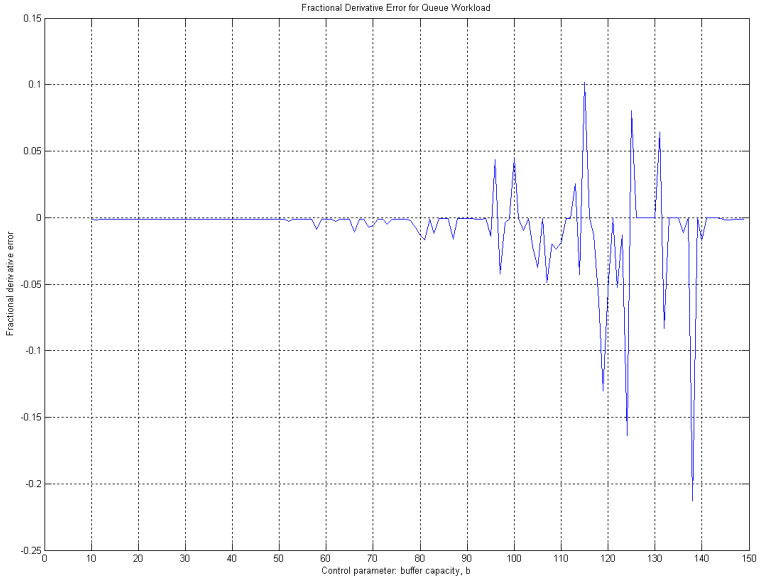


(a) IPA loss-derivative error (Matlab)

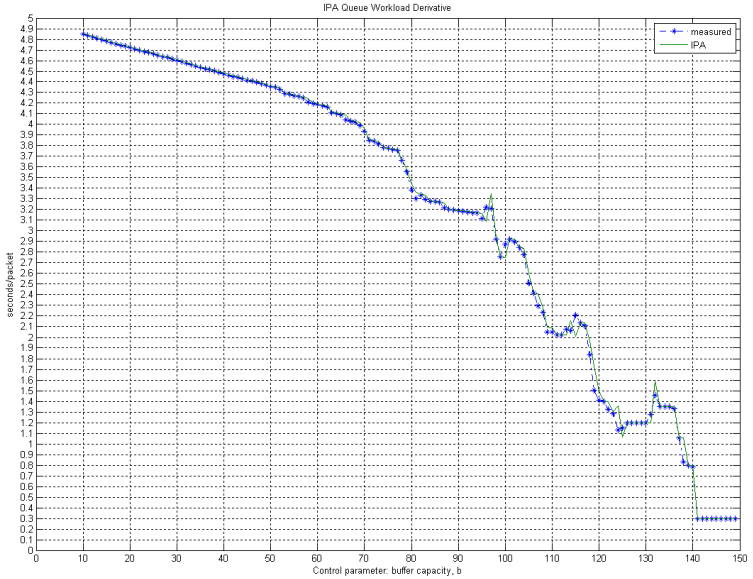


(b) IPA loss-derivative (Matlab)

Figure 4: IPA derivative results using Matlab



(c) IPA queue-workload-derivative error (Matlab)



(d) IPA queue-workload-derivative (Matlab)

Figure 4: IPA derivative results using Matlab

The stepsize or gain, $a_i = \frac{a_0}{i^\rho}$ where a_0 is the initial stepsize and $0.5 < \rho < 1$ is a constant that ensures convergence of the sequence $a_i, i = 1, 2, \ldots$. For this set of optimizations, $a_0 = 5, \rho = 0.6$. $\frac{dJ}{d\theta}$ is the IPA estimate of the derivative of the expected cost $E[J(\theta)]$ where $J(\theta) = \frac{w_L}{T} L_T + \frac{w_Q}{T} Q_T$. Therefore, $\frac{dJ}{d\theta} = \frac{w_L}{T} \frac{dL_T}{d\theta} + \frac{w_Q}{T} \frac{dQ_T}{d\theta}$. Here $w_L = 10$ and $w_Q = 70$. The optimizations were performed for different initial values of θ, (i.e. 20 packets, 70 packets and 200 packets). Figure 5 shows the graphical results of this optimization.

2.4 Simulations with loss-feedback constant, c, as the control variable θ

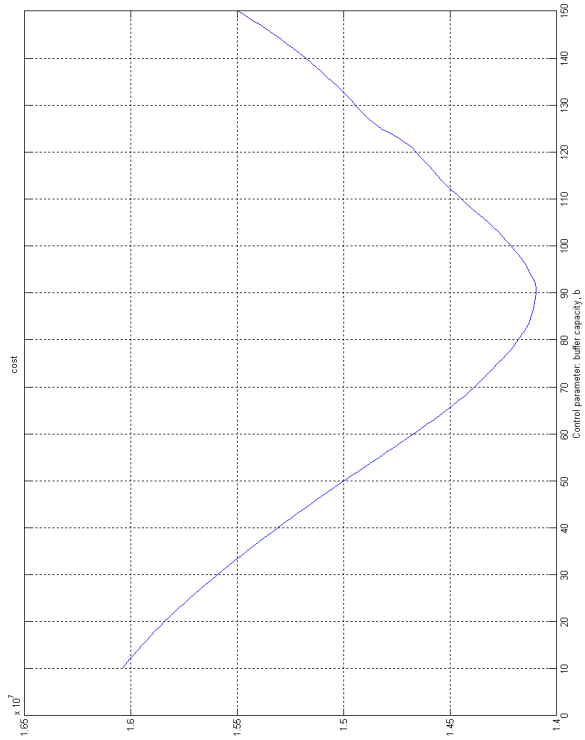
The same test configuration was used as in Figure 3. For both the error analysis and optimization, Matlab was used.

2.4.1 IPA derivative error-analysis

For the error analysis, the buffer capacity b of the first queue, Q_1, was held constant at 20 packets, and the loss-feedback constant, c, varied from 0 to 1.5 in increments of 0.01. The plot of the IPA fractional loss-derivative error versus θ is shown in Figure 6a. Here was also plotted the theoretical error value as from Eq. (32). It can be seen that the error values track very closely. Now, because the queue-workload derivative with respect to the loss-feedback constant is zero, the fractional derivative error could not be computed, however the measured derivative was still obtained (i.e. $\frac{Q_T(\theta+\Delta\theta)-Q_T(\theta)}{\Delta\theta}$) and plotted. See Figure 6b. For the sole purpose of verification of the derivative, the cost function was taken to be $L_T + kc$ where $k = 2000$ to ensure that convergence to a minimum occured during the optimization process[3]. The plot is shown in Figure 7a.

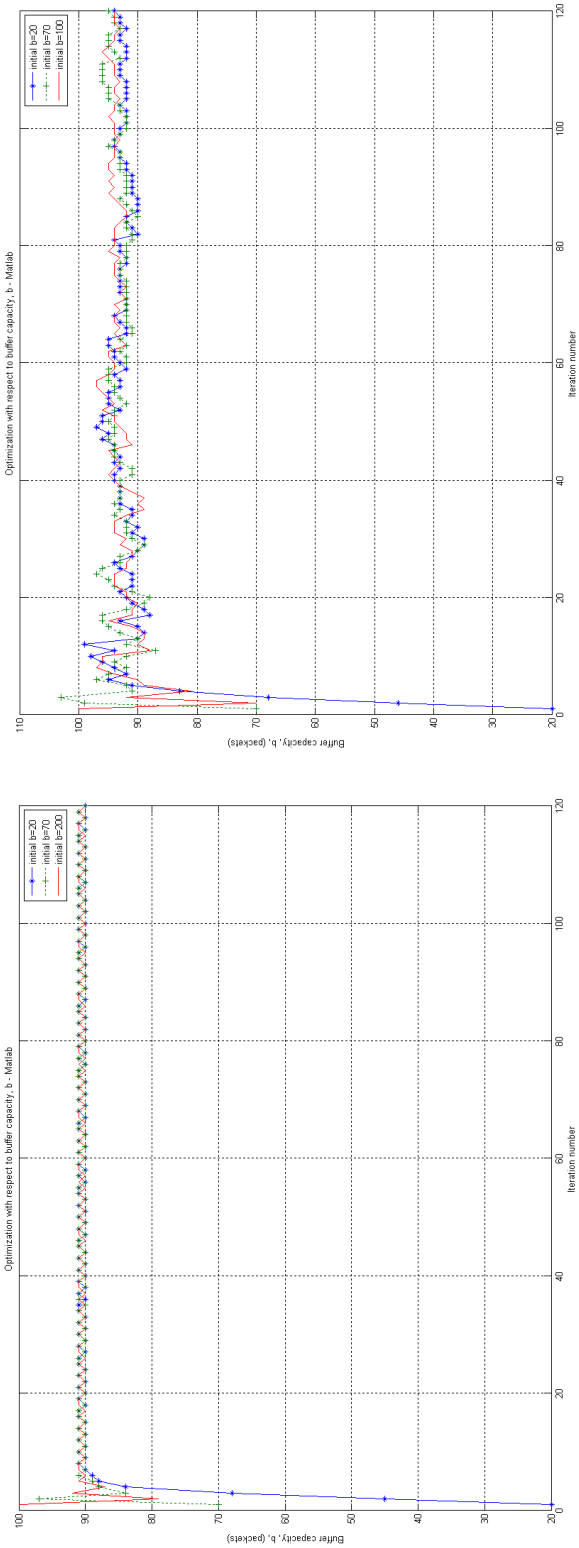
We also performed the alternative approach to analysis by using three sets of four adjacent values of the control variable, c, and three different seeds of the random number generator. See Table 2. The column, $\epsilon_{theory}(\%)$, is that error which is calculated in terms of θ and $\Delta\theta$ as in Eq. (32). We compare the value of $\epsilon_{theory}(\%)$ with the actual error, $\epsilon_L(\%)$. It can be seen that $\epsilon_L(\%)$ compares quite closely to $\epsilon_{theory}(\%)$.

[3]There is no justification for the term kc in reality



(a) Cost function($L_T + 7.0Q_T$)

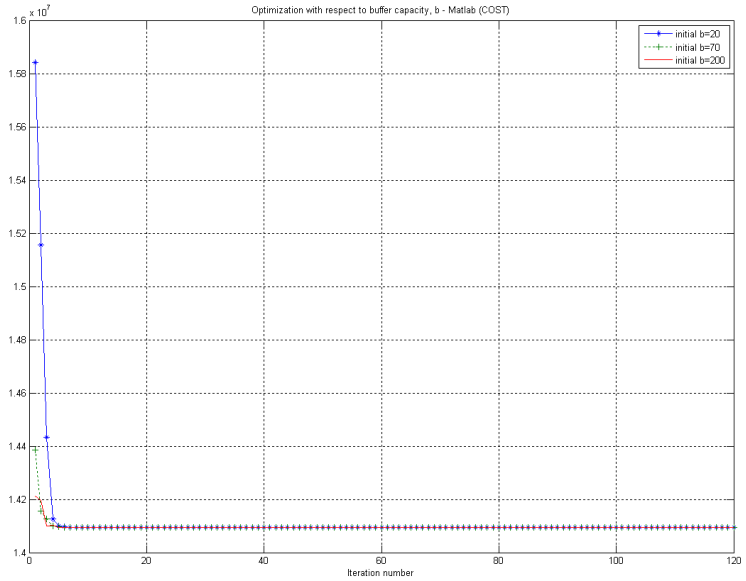
Figure 5: IPA optimization with respect to buffer capacity, b



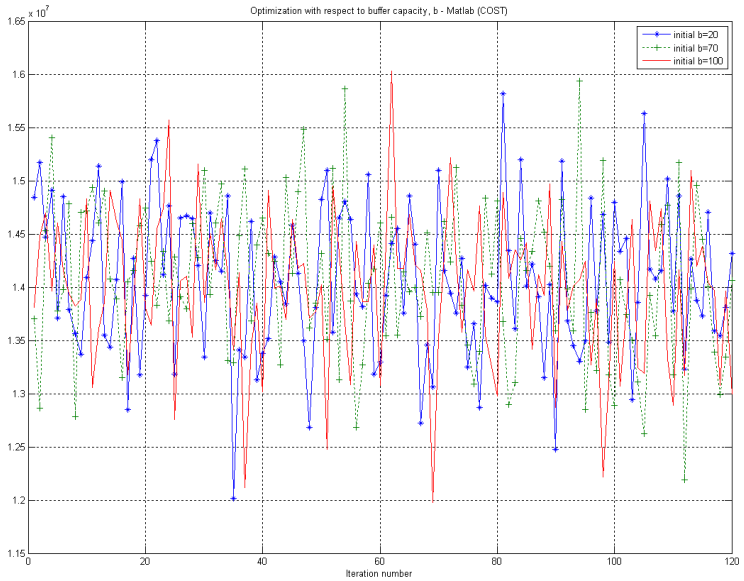
(b) Optimization with same seed

(c) Optimization with randomness

Figure 5: IPA optimization with respect to buffer capacity, b



(d) Cost progression with same seed

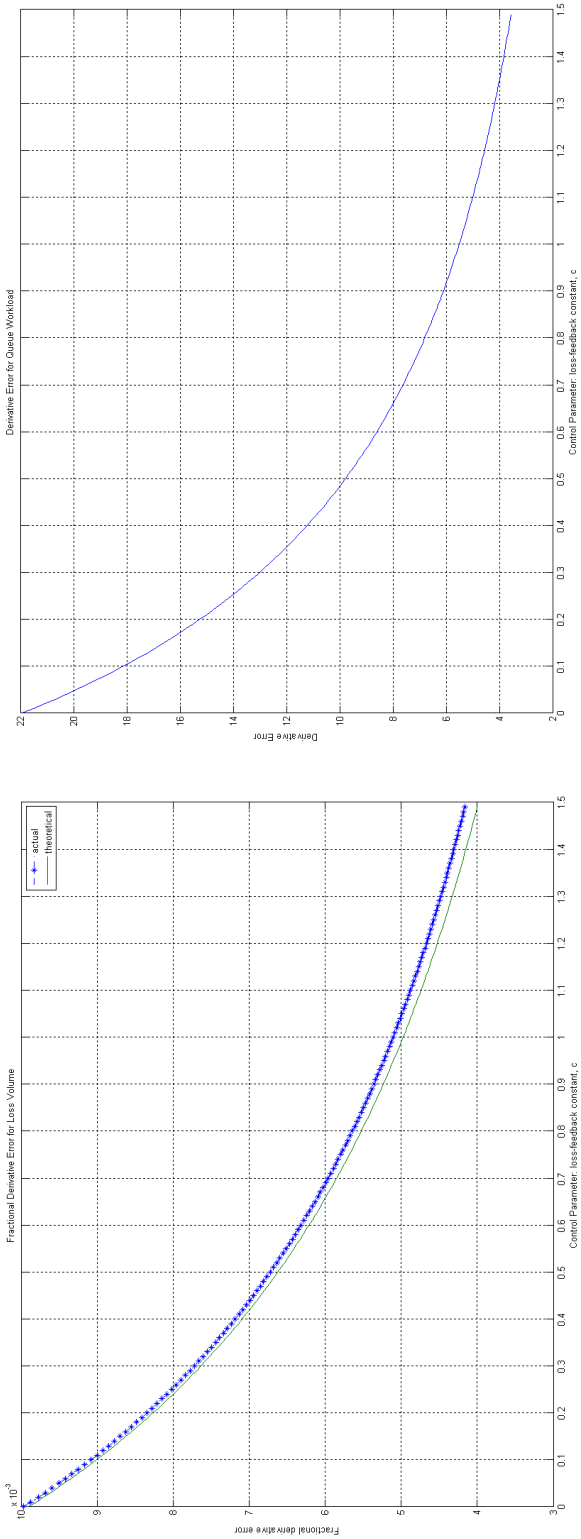


(e) Cost progression with randomness

Figure 5: IPA optimization with respect to buffer capacity, b

Table 2: IPA sensitivity analysis - the single-stage case $\theta \equiv c$

	seed=9736					
c	$\frac{dL}{d\theta}$	L(bits)	$\epsilon_L(\%)$	$\epsilon_{theory}(\%)$	$\frac{dQ}{d\theta}$	Q(bits-seconds)
0	-411138818.1	411138818.1	9.09682	9.09091	0	10310170.54
0.1	-339786355.6	373764991.2	8.33989	8.33333	0	10310219.22
0.2	-285516779.5	342620135.4	7.69951	7.69231	0	10310259.78
0.3	-243282150.8	316266796			0	10310294.1
0.7	-142269039.3	241857366.9	5.56599	5.55556	0	10310391.02
0.8	-126901295	228422331.1	5.27424	5.26316	0	10310408.52
0.9	-113895531.3	216401509.5	5.01173	5.00000	0	10310424.18
1	-102791385.1	205582770.2			0	10310438.27
1.3	-77726570.33	178771111.8	4.18099	4.16667	0	10310473.2
1.4	-71384762.03	171323428.9	4.01497	4.00000	0	10310482.9
1.5	-65788624.24	164471560.6	3.86178	3.84615	0	10310491.83
1.6	-60825676.59	158146759.1			0	10310500.06
	seed=1000					
c	$\frac{dL}{d\theta}$	L(bits)	$\epsilon_L(\%)$	$\epsilon_{theory}(\%)$	$\frac{dQ}{d\theta}$	Q(bits-seconds)
0	-411619412.2	411619412.2	9.09683	9.09091	0	10307547
0.1	-340183548.7	374201903.6	8.33990	8.33333	0	10307596
0.2	-285850538.7	343020646.4	7.69952	7.69231	0	10307636
0.3	-243566542.8	316636505.7			0	10307670
0.7	-142435357.9	242140108.4	5.56602	5.55556	0	10307767
0.8	-127049650	228689370.1	5.27427	5.26316	0	10307784
0.9	-114028683.6	216654498.9	5.01176	5.00000	0	10307800
1	-102911557.4	205823114.8			0	10307814
1.3	-77817443.26	178980119.5	4.18103	4.16667	0	10307849
1.4	-71468221.63	171523731.9	4.01501	4.00000	0	10307858
1.5	-65865542.13	164663855.3	3.86181	3.84615	0	10307867
1.6	-60896792.91	158331661.6			0	10307875
	seed=5586					
c	$\frac{dL}{d\theta}$	L(bits)	$\epsilon_L(\%)$	$\epsilon_{theory}(\%)$	$\frac{dQ}{d\theta}$	Q(bits-seconds)
0	-407007319.5	407007319.5	9.09688	9.09091	0	10291987
0.1	-336371895	370009084.5	8.33996	8.33333	0	10292035
0.2	-282647684.9	339177221.9	7.69958	7.69231	0	10292076
0.3	-240837479.1	313088722.8			0	10292110
0.7	-140839457.8	239427078.2	5.56610	5.55556	0	10292207
0.8	-125626143.9	226127059.1	5.27436	5.26316	0	10292224
0.9	-112751074.7	214227041.9	5.01186	5.00000	0	10292240
1	-101758513.3	203517026.5			0	10292254
1.3	-76945571.47	176974814.4	4.18114	4.16667	0	10292289
1.4	-70667490.7	169601977.7	4.01513	4.00000	0	10292299
1.5	-65127587.16	162818967.9	3.86194	3.84615	0	10292308
1.6	-60214510.82	156557728.1			0	10292316



(a) IPA loss-derivative error

(b) IPA queue-workload-derivative error

Figure 6: IPA derivative error with respect to the loss-feedback constant c

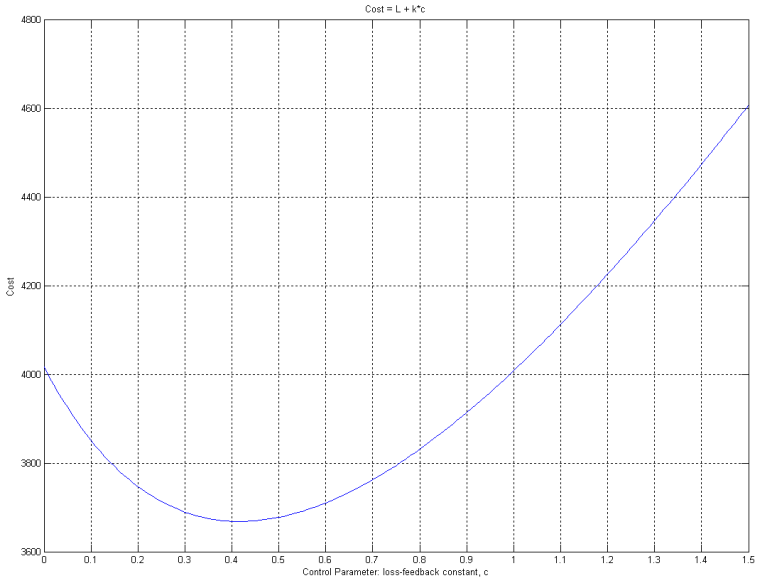
2.4.2 IPA optimization

Each optimization was carried out for 120 iterations, with each iteration being a run of length 10s. The control variable θ (i.e. the loss-feedback constant, c) was adjusted after each run and was used as the new loss-feedback constant in the next run. The algorithm for the adjustment in c was as follows:

$$\theta_{i+1} = \theta_i - a_i \frac{dJ}{d\theta}(\theta_i) \tag{33}$$

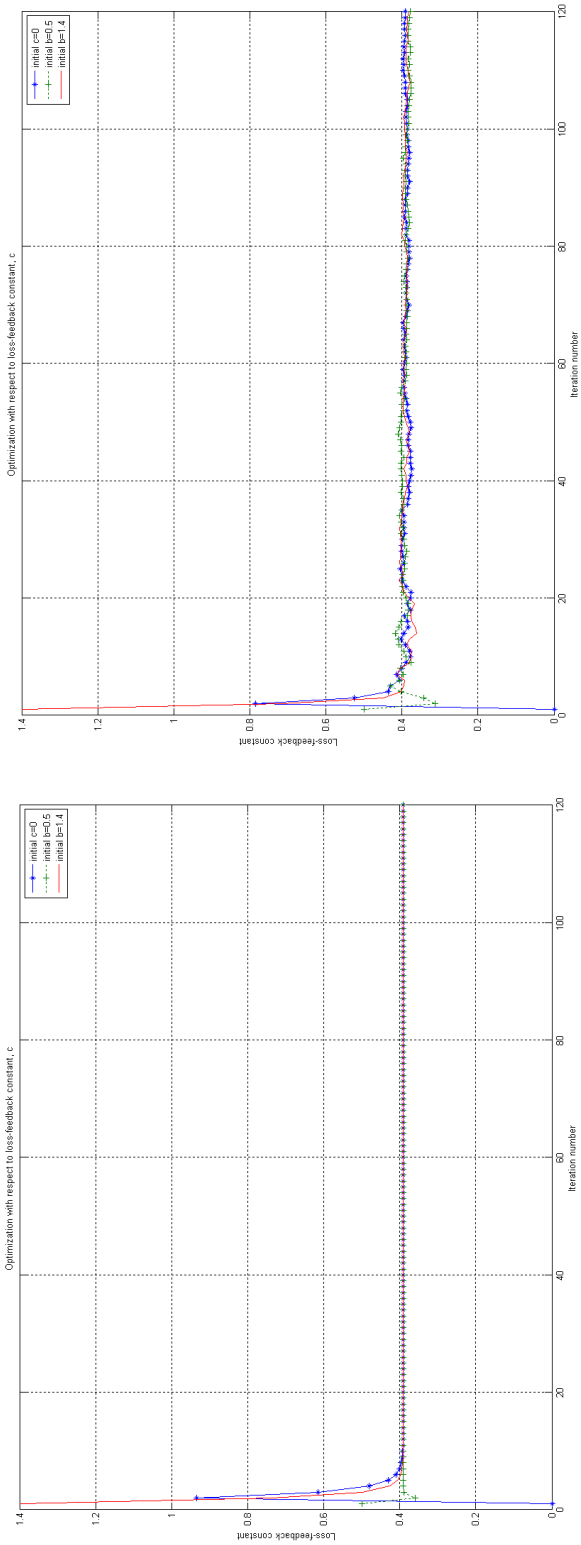
The stepsize or gain, $a_i = \frac{a_0}{i^\rho}$ where $a_0 = 0.0005, \rho = 0.6$. Also $\frac{dJ}{d\theta} = \frac{w_L}{T}\frac{dL_T}{d\theta} + k$. Here $w_L = 10$ and $k = 2000$.

Two cases were considered. For each run, the same seed was used for the random-number generator, therefore the the optimization should converge to the minimum of the cost function as in Figure 7a which was $c = 0.39$. The next case, has each run completely independent of each other (i.e. a random seed). Three different initial values of c were used, i.e., $c = \{0.0, 0.5, 1.4\}$. The results for the first case are seen in Figure 7b. It converges to $c = 0.39$. Figure 7c shows the results of the second case. There is still convergence in the presence of noise, but slightly away from the true minimum.



(a) Cost function

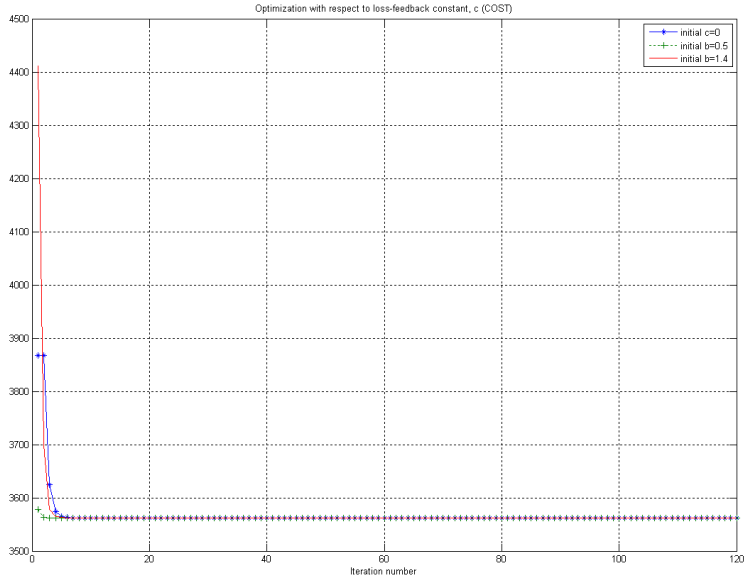
Figure 7: IPA optimization with respect to loss-feedback constant, c



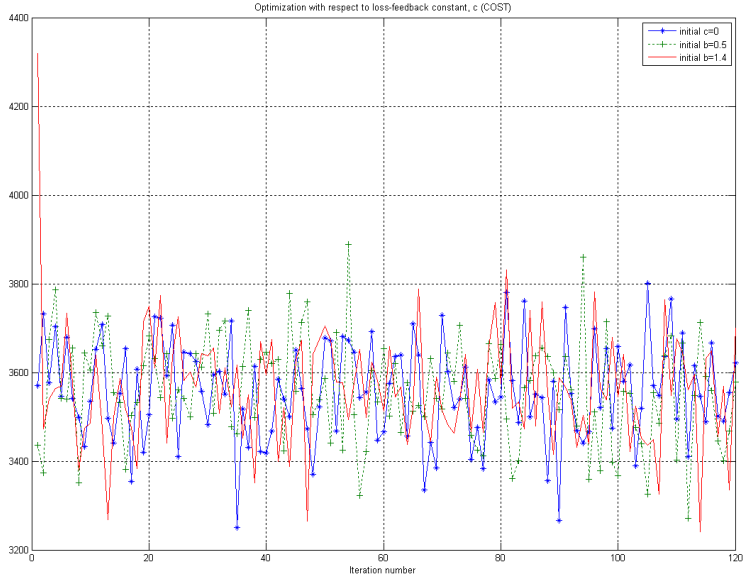
(b) Optimization with same seed

(c) Optimization with randomness

Figure 7: IPA optimization with respect to loss-feedback constant, c



(d) Cost progression with same seed



(e) Cost progression with randomness

Figure 7: IPA optimization with respect to loss-feedback constant, c

CHAPTER III

SINGLE-STAGE FLUID QUEUE WITH LOSS FEEDBACK AND NON-RESPONSIVE COMPETING TRAFFIC

We now consider the case when a non-responsive stochastic flow of rate $\sigma_2(t)$ competes with that of the IPA-controlled source. The rate of the latter is designated as $\sigma_1(t)$. This new scenario is depicted in Figure 8. Again the loss-feedback constant is c, the buffer capacity, b, the buffer occupancy, $x(\theta;t)$, the service rate, $\beta(t)$, the outflow rate, $\delta(\theta;t)$ and the IPA time-horizon, the constant T such that $T > 0$ and $t \in [0, T]$. The total loss rate at the queue is denoted as $\gamma(\theta;t) = \gamma_1(\theta;t) + \gamma_2(\theta;t)$ where $\gamma_1(\theta;t)$ is the loss-rate of the IPA-controlled flow, and $\gamma_2(\theta;t)$ is that of the non-responsive flow.

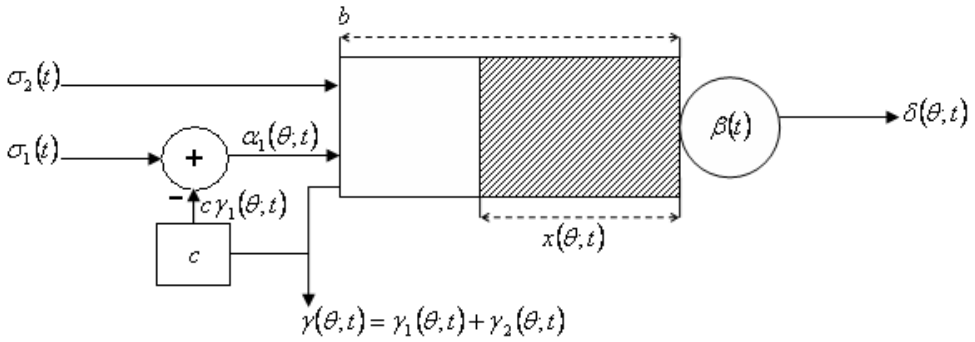


Figure 8: Stochastic fluid model with instantaneous additive-loss feedback for the single stage case with a competing non-responsive flow

Define:

$$A(\theta;t) = \sigma_2(t) + \alpha_1(\theta;t) - \beta(t) \tag{34}$$

then:

$$\frac{dx}{dt} = \begin{cases} 0 & \text{if } x(\theta;t) = 0 \text{ and } A(\theta;t) < 0 \\ 0 & \text{if } x(\theta;t) = b \text{ and } A(\theta;t) > 0 \\ A(\theta;t) & \text{otherwise} \end{cases} \tag{35}$$

also:

$$\gamma(\theta;t) = \begin{cases} A(\theta;t) & \text{if } x(\theta;t) = b \\ 0 & \text{otherwise} \end{cases} \tag{36}$$

Now:

$$\alpha_1(\theta;t) \;=\; \sigma_1(t) - c\gamma_1(\theta;t) \tag{37}$$

where

$$\gamma_1(\theta;t) \;=\; \tfrac{\alpha_1(\theta;t)}{\alpha_1(\theta;t)+\sigma_2(t)}\gamma(\theta;t) \tag{38}$$

$$\begin{aligned} \Rightarrow \alpha_1(\theta;t) &= \begin{cases} f(\sigma_1(t),\sigma_2(t),\beta(t),c) & \text{if } x(\theta;t) = b \\ \sigma_1(t) & \text{otherwise} \end{cases} \\ \Rightarrow A(\theta;t) &= \begin{cases} \sigma_2(t) + f(\sigma_1(t),\sigma_2(t),\beta(t),c) - \beta(t) & \text{if } x(\theta;t) = b \\ \sigma_2(t) + \sigma_1(t) - \beta(t) & \text{otherwise} \end{cases} \end{aligned} \tag{39}$$

where

$$f(\sigma_1(t),\sigma_2(t),\beta(t),c) \;\equiv\; -\tfrac{\sigma_2(t)}{2} + \tfrac{\sigma_1(t)+c\beta(t)}{2(1+c)} + \sqrt{\left(\tfrac{\sigma_2(t)}{2} - \tfrac{\sigma_1(t)+c\beta(t)}{2(1+c)}\right)^2 + \sigma_1(t)\sigma_2(t)} \tag{40}$$

The derivation of Eq. (40) can be found in Appendix A.

Using the relation in Eq. (39), Eq. (35) and Eq. (36) can be rewritten as:

$$\frac{dx}{dt} = \begin{cases} 0 & \text{if } x(\theta;t) = 0 \text{ and } \sigma_2(t) + \sigma_1(t) - \beta(t) < 0 \\ 0 & \text{if } x(\theta;t) = b \text{ and } \sigma_2(t) + f(\sigma_1(t),\sigma_2(t),\beta(t),c) - \beta(t) > 0 \\ \sigma_2(t) + \sigma_1(t) - \beta(t) & \text{otherwise} \end{cases} \tag{41}$$

$$\gamma(\theta;t) = \begin{cases} \sigma_2(t) + f(\sigma_1(t),\sigma_2(t),\beta(t),c) - \beta(t) & \text{if } x(\theta;t) = b \\ 0 & \text{otherwise} \end{cases} \tag{42}$$

Consider the buffer-occupancy process in Figure 9. Let $\tau_i(\theta)$ be the time of the beginning of the i-th full period and let $\varsigma_i(\theta)$ be the time of the end of the i-th full period. Then the total loss volume, denoted as $L_T(\theta)$, is given by:

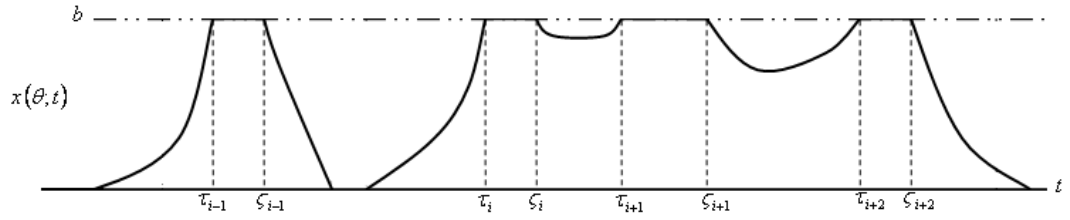


Figure 9: Trajectory of buffer-occupancy for single-stage case with competing flow

$$L_T(\theta) = \sum_{i=1}^{K} L_i(\theta) \tag{43}$$

where K is the number of full-periods in the interval $[0, T]$ and

$$\begin{aligned} L_i(\theta) &= \int_{\tau_i(\theta)}^{\varsigma_i(\theta)} \gamma(\theta; t)\, dt \\ &= \int_{\tau_i(\theta)}^{\varsigma_i(\theta)} [\sigma_2(t) + f(\sigma_1(t), \sigma_2(t), \beta(t), c) - \beta(t)]\, dt \end{aligned} \tag{44}$$

The loss derivative is then given by

$$\frac{dL_T(\theta)}{d\theta} = \sum_{i=1}^{K} \frac{dL_i(\theta)}{d\theta} \tag{45}$$

The same definition of events and assumptions that were made in the single-stage case without competing flows will also be used here.

3.1 Buffer capacity, b, as the control parameter, θ

$$\begin{aligned} \frac{dL_i(\theta)}{d\theta} &= \frac{d}{d\theta}\left[\int_{\tau_i(\theta)}^{\varsigma_i(\theta)} (\sigma_2(t) + f(\sigma_1(t), \sigma_2(t), \beta(t), c) - \beta(t))\, dt\right] \\ &= \int_{\tau_i(\theta)}^{\varsigma_i(\theta)} \frac{d}{d\theta}[\sigma_2(t) + f(\sigma_1(t), \sigma_2(t), \beta(t), c) - \beta(t)]\, dt \\ &\quad + (\sigma_2(\varsigma_i(\theta)^-) + f(\sigma_1(\varsigma_i(\theta)^-), \sigma_2(\varsigma_i(\theta)^-), \beta(\varsigma_i(\theta)^-), c) - \beta(\varsigma_i(\theta)^-))\frac{d\varsigma_i(\theta)}{d\theta} \\ &\quad - (\sigma_2(\tau_i(\theta)^+) + f(\sigma_1(\tau_i(\theta)^+), \sigma_2(\tau_i(\theta)^+), \beta(\tau_i(\theta)^+), c) - \beta(\tau_i(\theta)^+))\frac{d\tau_i(\theta)}{d\theta} \end{aligned} \tag{46}$$

Now $\sigma_1(t)$, $\sigma_2(t)$ and $\beta(t)$ are independent of θ, then $f(\sigma_1(t), \sigma_2(t), \beta(t), c)$ is also independent of θ within the interval $(\tau_i(\theta), \varsigma_i(\theta))$ so that $\int_{\tau_i(\theta)}^{\varsigma_i(\theta)} \frac{d}{d\theta}\sigma_2(t) + f(\sigma_1(t), \sigma_2(t), \beta(t), c) - \beta(t)]\, dt = 0$

Consider, $t = \varsigma_i(\theta)$, the time when the queue ceases to be full. For the discontinuous case, there is a jump in $A(\theta; t)$ i.e. $A(\theta; \varsigma_i(\theta)^-) > A(\theta; \varsigma_i(\theta)^+)$. This implies that $\sigma_2(\varsigma_i(\theta)^-) + f(\sigma_1(\varsigma_i(\theta)^-), \sigma_2(\varsigma_i(\theta)^-), \beta(\varsigma_i(\theta)^-), c) - \beta(\varsigma_i(\theta)^-) > \sigma_2(\varsigma_i(\theta)^+) + \sigma_1(\varsigma_i(\theta)^+) -$

$\beta(\varsigma_i(\theta)^+)$. Suppose that there are no exogenous changes in the processes $\sigma_1(t)$, $\sigma_2(t)$ and $\beta(t)$ at $t = \varsigma_i(\theta)$, then $f(\sigma_1(\varsigma_i(\theta)), \sigma_2(\varsigma_i(\theta)), \beta(\varsigma_i(\theta)), c) > \sigma_1(\varsigma_i(\theta))$. But this cannot be since for the same values of $\sigma_1(t)$, $\sigma_2(t)$ and $\beta(t)$ at $t = \varsigma_i(\theta)$ $\alpha_1(\varsigma_i(\theta)^-) < \alpha_1(\varsigma_i(\theta)^+) \Rightarrow f(\sigma_1(\varsigma_i(\theta)), \sigma_2(\varsigma_i(\theta)), \beta(\varsigma_i(\theta)), c) < \sigma_1(\varsigma_i(\theta))$. Therefore, for there to be a negative jump in $A(\theta; t)$, it must be due to an exogenous event in $\sigma_1(t)$, $\sigma_2(t)$ or $\beta(t)$ at $t = \varsigma_i(\theta)$. As a result, $\frac{d\varsigma_i(\theta)}{d\theta} = 0$.

In the continuous case, $A(\theta; t) = 0$ at $t = \varsigma_i(\theta)$ which implies that

$$f(\sigma_1(\varsigma_i(\theta)), \sigma_2(\varsigma_i(\theta)), \beta(\varsigma_i(\theta)), c) = \sigma_1(\varsigma_i(\theta)) = \sigma_2(\varsigma_i(\theta)) - \beta(\varsigma_i(\theta)) \tag{47}$$

(which is possible) so that $A(\theta; \varsigma_i(\theta))\frac{d\varsigma_i(\theta)}{d\theta} = 0$.

Eq. (46) then becomes:

$$\frac{dL_i(\theta)}{d\theta} = -(\sigma_2(\tau_i(\theta)^+) + f(\sigma_1(\tau_i(\theta)^+), \sigma_2(\tau_i(\theta)^+), \beta(\tau_i(\theta)^+), c) - \beta(\tau_i(\theta)^+))\frac{d\tau_i(\theta)}{d\theta} \tag{48}$$

Consider the non-boundary period (NBP) that precedes the full-period. That NBP may have itself been preceded by an empty period (EF) or a full-period (FF).

$$\begin{aligned} x(\theta; t)|_{\varsigma_{i-1}(\theta)}^{\tau_i(\theta)} &= x(\theta; \tau_i(\theta)) - x(\theta; \varsigma_{i-1}(\theta)) = \begin{cases} \theta & \text{if } NBP \text{ is } EF \\ 0 & \text{if } NBP \text{ is } FF \end{cases} \\ &= \int_{\varsigma_{i-1}(\theta)}^{\tau_i(\theta)} \frac{dx(\theta,t)}{dt}\, dt \\ &= \int_{\varsigma_{i-1}(\theta)}^{\tau_i(\theta)} (\sigma_2(t) + \sigma_1(t) - \beta(t))\, dt \end{aligned} \tag{49}$$

Taking derivatives of Eq. (49) with respect to θ:

$$(\sigma_2(\tau_i(\theta)^-) + \sigma_1(\tau_i(\theta)^-) - \beta(\tau_i(\theta)^-))\frac{d\tau_i(\theta)}{d\theta} = \begin{cases} 1 & \text{if } NBP \text{ is } EF \\ 0 & \text{if } NBP \text{ is } FF \end{cases} \tag{50}$$

When the queue becomes full at $t = \tau_i(\theta)$ corresponds to an endogenous event at which point it is assumed that there can be no exogenous event occurring. Therefore $(\sigma_2(\tau_i(\theta)^-) + \sigma_1(\tau_i(\theta)^-) - \beta(\tau_i(\theta)^-)) = (\sigma_2(\tau_i(\theta)^+) + \sigma_1(\tau_i(\theta)^+) - \beta(\tau_i(\theta)^+)) = (\sigma_2(\tau_i(\theta)) + \sigma_1(\tau_i(\theta)) - \beta(\tau_i(\theta)))$.

Eq. (50) then becomes:

$$(\sigma_2(\tau_i(\theta)) + \sigma_1(\tau_i(\theta)) - \beta(\tau_i(\theta)))\frac{d\tau_i(\theta)}{d\theta} = \begin{cases} 1 & \text{if } NBP \text{ is } EF \\ 0 & \text{if } NBP \text{ is } FF \end{cases} \tag{51}$$

Substitute Eq. (51) into Eq. (48):

$$\frac{dL_i(\theta)}{d\theta} = \begin{cases} -\frac{(\sigma_2(\tau_i(\theta))+f(\sigma_1(\tau_i(\theta)),\sigma_2(\tau_i(\theta)),\beta(\tau_i(\theta)),c)-\beta(\tau_i(\theta)))}{(\sigma_2(\tau_i(\theta))+\sigma_1(\tau_i(\theta))-\beta(\tau_i(\theta)))} & \text{if preceding } NBP \text{ is } EF \\ 0 & \text{if preceding } NBP \text{ is } FF \end{cases} \tag{52}$$

3.2 An alternative view

We can express the inflow rate to the queue, $\alpha_1(\theta;t)$, as

$$\alpha(\theta;t) = \sigma_1(t) - g(\gamma(\theta;t)) \tag{53}$$

where $g \geq 0, g^{'} \geq 0$. Therefore

$$\frac{dL_i(\theta)}{d\theta} = \begin{cases} 1 - \frac{g(\gamma_1(\theta;\tau_i(\theta))}{(\sigma_2(\tau_i(\theta))+\sigma_1(\tau_i(\theta))-\beta(\tau_i(\theta)))} & \text{if preceding } NBP \text{ is } EF \\ 0 & \text{if preceding } NBP \text{ is } FF \end{cases} \tag{54}$$

Because the loss-feedback is felt only when the queue is in a full-period, the IPA derivative for the queue workload is the same as that of the single-stage case with no loss-feedback.

3.3 Feedback constant, c, as the control parameter, θ

$$\begin{aligned} \frac{dL_i(\theta)}{d\theta} &= \frac{d}{d\theta}\left[\int_{\tau_i(\theta)}^{\varsigma_i(\theta)}(\sigma_2(t) + f(\sigma_1(t),\sigma_2(t),\beta(t),c) - \beta(t))\,dt\right] \\ &= \int_{\tau_i(\theta)}^{\varsigma_i(\theta)}\frac{d}{d\theta}[\sigma_2(t) + f(\sigma_1(t),\sigma_2(t),\beta(t),c) - \beta(t)]\,dt \\ &\quad +(\sigma_2(\varsigma_i(\theta)^-) + f(\sigma_1(\varsigma_i(\theta)^-),\sigma_2(\varsigma_i(\theta)^-),\beta(\varsigma_i(\theta)^-),c) - \beta(\varsigma_i(\theta)^-))\frac{d\varsigma_i(\theta)}{d\theta} \\ &\quad -(\sigma_2(\tau_i(\theta)^+) + f(\sigma_1(\tau_i(\theta)^+),\sigma_2(\tau_i(\theta)^+),\beta(\tau_i(\theta)^+),c) - \beta(\tau_i(\theta)^+))\frac{d\tau_i(\theta)}{d\theta} \end{aligned} \tag{55}$$

Now, using the same line of argument as in the previous section, it is found that:

$$\begin{aligned} (\sigma_2(\varsigma_i(\theta)^-) + f(\sigma_1(\varsigma_i(\theta)^-),\sigma_2(\varsigma_i(\theta)^-),\beta(\varsigma_i(\theta)^-),c) - \beta(\varsigma_i(\theta)^-))\frac{d\varsigma_i(\theta)}{d\theta} &= 0 \\ (\sigma_2(\tau_i(\theta)^+) + f(\sigma_1(\tau_i(\theta)^+),\sigma_2(\tau_i(\theta)^+),\beta(\tau_i(\theta)^+),c) - \beta(\tau_i(\theta)^+))\frac{d\tau_i(\theta)}{d\theta} &= 0 \end{aligned} \tag{56}$$

so that

$$\frac{dL_i(\theta)}{d\theta} = \int_{\tau_i(\theta)}^{\varsigma_i(\theta)}\frac{d}{d\theta}[f(\sigma_1(t),\sigma_2(t),\beta(t),c)]\,dt \tag{57}$$

where

$$\frac{d}{d\theta}[f(\sigma_1(t),\sigma_2(t),\beta(t),c)] \quad = \quad -\frac{\sigma_1(t)-\beta(t)}{2(1+c)}\left[1+\frac{\frac{\sigma_2(t)}{2}-\frac{\sigma_1(t)+c\beta(t)}{1+c}}{\sqrt{\left(\frac{\sigma_2(t)}{2}-\frac{\sigma_1(t)+c\beta(t)}{1+c}\right)^2+\sigma_1(t)\sigma_2(t)}}\right] \tag{58}$$

The queue workload is independent of changes in c, the feedback constant, therefore $\frac{dQ_T(\theta)}{d\theta} = 0$ for all θ.

3.4 An approximation

Let $\Gamma(t) \equiv \frac{\sigma_2(t)}{\sigma_1(t)+\sigma_2(t)}$, i.e. the instantaneous fraction of the total source rates that belongs to the non-responsive flow. If it is assumed that $0 < c \ll 1$ so that $\frac{\sigma_2(t)}{\alpha_1(t)+\sigma_2(t)} \approx \frac{\sigma_2(t)}{\sigma_1(t)+\sigma_2(t)}$ then substitute into Eq. (37) and Eq. (39) so that $\alpha_1(\theta;t) = \sigma_1(t) - c(1-\Gamma(t))\gamma(\theta;t)$ and $\gamma(\theta;t) = \frac{\sigma_2(t)+\sigma_1(t)-\beta(t)}{1+c(1-\Gamma(t))}$.

When the buffer capacity, b, is the control parameter θ:

$$\frac{dL_i(\theta)}{d\theta} = \begin{cases} -\frac{1}{1+c(1-\Gamma(\tau_i(\theta)))} & \text{if preceding } NBP \text{ is } EF \\ 0 & \text{if preceding } NBP \text{ is } FF \end{cases} \tag{59}$$

When the loss-feedback constant, c, is the control parameter θ:

$$\frac{dL_i(\theta)}{d\theta} \quad = \quad -\frac{1-\Gamma(\tau_i(\theta))}{1+c(1-\Gamma(\tau_i(\theta)))}L_i \tag{60}$$

3.5 Simulations with buffer capacity, b, as the control variable θ

To verify the accuracy of the IPA gradient estimators derived for the single-stage fluid model with loss feedback and competing non-responsive traffic, the test configuration as shown in Figure 10 was used.

The first queue, Q_1, calculates the IPA loss and queue-workload derivatives. Its service rate, $\beta_1(t)$, is a random process, uniformly distributed between 6.75 Mbps and 11.25 Mbps. Its average service rate is 9 Mbps. The time interval between changes in the magnitude of $\beta_1(t)$ is a constant at 75 ms. The second queue, Q_2, is an ordinary first-in-first-out droptail queue with a constant service rate of 20 Mbps. There is no queueing at the second queue, since $\beta_1(t) < \beta_2(t)$ for all $t \in [0,T]$ where $T = 10$ seconds. The average input-rate,

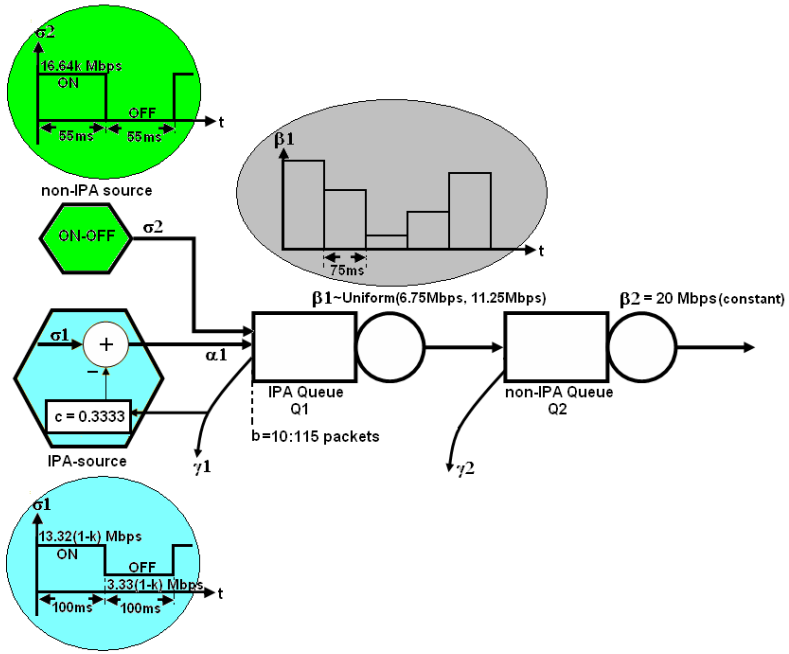


Figure 10: Test configuration for the single-stage case with a non-responsive competing flow

$\lambda = E[\sigma_1] + E[\sigma_2]$, into the first queue was held at 8.31769 Mbps (taking into account packet-header adjustments). Let $k \equiv \frac{E[\sigma_2]}{E[\sigma_1]+E[\sigma_2]}$ Therefore

$$\begin{aligned} x &= (1-k) \times \lambda \\ y &= k \times \lambda \end{aligned} \tag{61}$$

For each IPA-controlled source, let p be the "on-rate", and q, the "off-rate". p and q are calculated as follows:

$$\begin{aligned} p &= \frac{x}{d+(1-d)r} \\ q &= rp \end{aligned} \tag{62}$$

where

$$\begin{aligned} d &\equiv \text{duty-cycle} = 0.5 \\ r &\equiv \text{ratio of off- to on-rate} = 0.25 \end{aligned}$$

The total (on- + off-)period on average was $200ms$.

For the non-responsive source, let a be the "on-rate", and b, the "off-rate". a and b are calculated as follows:

$$\begin{aligned} a &= \frac{y}{d} \\ b &= 0 \end{aligned} \tag{63}$$

where

$$d \equiv \text{duty-cycle} = 0.5$$

The total (on- + off-)period on average was $110ms$.

To summarize: $\sigma_1(t)$ switches between either of two levels, $13.31 \times (1-k)$ Mbps ("ON"-rate) and $3.32 \times (1-k)$ Mbps ("OFF"-rate) every 100 ms or so. Its average is $8.32 \times (1-k)$ Mbps. $\sigma_2(t)$ oscillates between $16.64 \times k$ Mbps and 0 Mbps every 55 ms or so, for an average of $8.32 \times k$ Mbps.

The loss-feedback constant, c is 0.3333. The packet size is 554 bytes (512 bytes for the payload, and 42 bytes for the header).

3.5.1 IPA derivative error-analysis

The error-analysis was carried out using Matlab. The buffer capacity of the Q_1, b, was incremented by 1 packet at a time (i.e. $\Delta\theta = 1$) from 10 packets to 90 packets. The same seed for the random-number generator was used for each value (i.e iterate) of the control variable. At each iterate, the error in the IPA loss-derivative ($\frac{dL_T}{d\theta}$)was calculated as in Eq. (64). The error in the IPA queue-workload derivative was calculated in a similar way. See Eq. (65) where Q_T is in packet-seconds. The cost funtion was also calculated for each value of the control variable. The cost function was taken as $\frac{w_L}{T}L_T + \frac{w_Q}{T}Q_T$. Here the weights were chosen to be $w_L = \frac{1.0}{554\times 8}$ and $w_L = \frac{8.35}{554\times 8}$.

$$\epsilon_L = \frac{\frac{L_T(\theta+\Delta\theta)-L_T(\theta)}{\Delta\theta} - \frac{dL_T(\theta)}{d\theta}}{\left|\frac{dL_T(\theta)}{d\theta}\right|} \tag{64}$$

$$\epsilon_Q = \frac{\frac{Q_T(\theta+\Delta\theta)-Q_T(\theta)}{\Delta\theta} - \frac{dQ_T(\theta)}{d\theta}}{\left|\frac{dQ_T(\theta)}{d\theta}\right|} \tag{65}$$

The error analysis was carried out for $k = 10\%, c = 0.3333$, $k = 10\%, c = 0.05$, $k = 25\%, c = 0.3333$ and $k = 25\%, c = 0.05$. The approximated IPA loss-derivative in Eq. (60) was plotted against the control variable θ on the same axes as the exact IPA loss-derivative (as in Eq. (52)) and the measured loss-derivative, $\frac{L_T(\theta+\Delta\theta)-L_T(\theta)}{\Delta\theta}$. This was to see how well the approximation holds for different values of the feedback constant, c, and the proportion, k, of the average source-rate that pertains to the non-responsive competing flow. See Figure 12. For completeness, a plot of the measured queue-workload-derivative with the IPA queue-workload derivative is shown in Figure 11c.

The IPA fractional loss-derivative error (when $k = 10\%, c = 0.3333$) was plotted against θ and is shown in Figure 11a. Also, the IPA fractional queue-workload derivative error for the same values of k and c was also plotted against θ and is shown in Figure 11b.

The alternative approach to error analysis was performed by using three sets of three adjacent values of the control variable, b for two different values of k, i.e., $\{10\%, 25\%\}$, and

three different seeds of the random number generator. Also $c = 0.3333$. (See Table 3 and Table 4 .)

For the group $b = \{29, 30, 31, 32\}$, the magnitude of the percentage error in the loss derivative was, for the most part, less than 0.05% when $k = 10\%$ and less than 0.20% when $k = 25\%$. The magnitude of the percentage error in queue-workload derivative was, for the most part, less than 0.2% when $k = 10\%$ and less than 0.40% when $k = 25\%$. For the group $b = \{50, 51, 52, 53\}$, the magnitude of the percentage error in the loss derivative was no greater than 0.575% when $k = 10\%$ and 1.45% when $k = 25\%$. The magnitude of the percentage error in queue-workload derivative was no greater than 0.45% when $k = 10\%$ and 1.14% when $k = 25\%$. For the group $b = \{99, 100, 101, 102\}$, the magnitude of the percentage error in the loss derivative did not exceed 3% when $k = 10\%$ and was just over 5.0% when $k = 25\%$. The magnitude of the percentage error in queue-workload derivative was no greater than 2.0% when $k = 10\%$ and less than 5.0% when $k = 25\%$.

3.5.2 IPA optimization

Each optimization was carried out for 120 iterations, with each iteration being a run of length 10s. The control variable θ (i.e. the buffer-limit in Q_1) was adjusted after each run and was used as the new buffer-limit in the next run. The algorithm for the adjustment in b was the same as that in the single-stage case (i.e., $a_0 = 5, \rho = 0.6$ and $\frac{dJ}{d\theta} = \frac{w_L}{T}\frac{dL_T}{d\theta} + \frac{w_Q}{T}\frac{dQ_T}{d\theta}$). However, $w_L = 10$ and $w_Q = 83.5$. So that

$$\theta_{i+1} = \theta_i - a_i \frac{dJ}{d\theta}(\theta_i) + \Delta \tag{66}$$

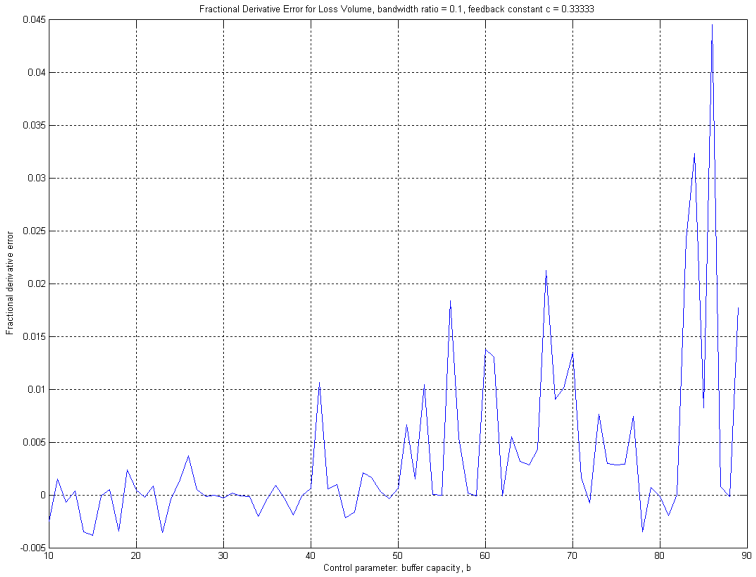
- if $|\lceil\theta_{i+1}\rceil - \theta_{i+1}| \leq |\lfloor\theta_{i+1}\rfloor - \theta_{i+1}|$
  - $\diamond$ $\Delta = \theta_{i+1} - \lceil\theta_{i+1}\rceil$
  - $\diamond$ $\theta_{i+1} = \lceil\theta_{i+1}\rceil$
- else
  - $\diamond$ $\Delta = \theta_{i+1} - \lfloor\theta_{i+1}\rfloor$
  - $\diamond$ $\theta_{i+1} = \lfloor\theta_{i+1}\rfloor$

Table 3: IPA sensitivity analysis - the single-stage with competing flow $\theta \equiv b$, $k = 10\%$

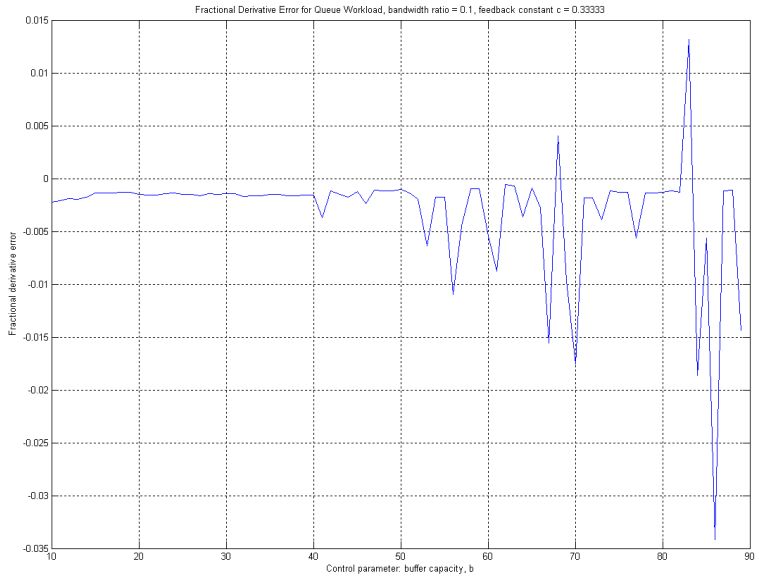
	seed=1000					
b (pkts)	$\frac{dL}{d\theta}$	L(bits)	$\epsilon_L(\%)$	$\frac{dQ}{d\theta}$	Q(bits-seconds)	$\epsilon_Q(\%)$
29	-989.141	247897622.6	-0.02415	107.1278	14517293	-0.18156
30	-989.709	243512689.3	-0.03604	106.7405	14991221	-0.17724
31	-989.775	239124718.3	-0.03017	106.3622	15463456	-0.18169
32	-990.105	234736713.9		105.976	15933997	
50	-936.106	157090648.8	0.57183	97.8407	24089964	-0.38610
51	-925.825	152965553.2	0.53350	97.08624	24521920	-0.34729
52	-917.95	148884188.8	0.49177	96.46734	24950712	-0.34225
53	-910.15	144835840.1		95.8547	25376792	
99	-265.978	21802069.23	2.16221	55.27601	40770899	-0.40302
100	-253.611	20648743.08	1.46547	55.03843	41014895	-0.19029
101	-245.789	19541212.32	2.13141	54.91395	41258361	-1.63708
102	-235.362	18475094.8		53.01358	41497755	
	seed=5586					
b (pkts)	$\frac{dL}{d\theta}$	L(bits)	$\epsilon_L(\%)$	$\frac{dQ}{d\theta}$	Q(bits-seconds)	$\epsilon_Q(\%)$
29	-990.161	244224239.4	0.02061	106.8419	14489194	-0.18944
30	-990.194	239836750	-0.00714	106.4369	14961820	-0.19138
31	-990.381	235447897.4	-0.00772	106.0274	15432646	-0.19293
32	-990.311	231058190.4		105.6184	15901652	
50	-928.705	153314375.6	0.57515	96.07298	23998784	-0.42917
51	-919.705	149222030	0.31977	95.32756	24422752	-0.29374
52	-913.138	145158931.8	0.49702	94.74382	24844003	-0.45529
53	-904.298	141132018		93.9117	25261995	
99	-256.211	20205770.76	2.36293	49.21834	39961266	-1.00988
100	-243.416	19097077.42	1.34868	48.41548	40177199	-0.71032
101	-237.683	18032806.91	1.29811	48.28101	40390252	-0.13063
102	-228.149	16993070.7		47.20017	40603954	
	seed=9736					
b (pkts)	$\frac{dL}{d\theta}$	L(bits)	$\epsilon_L(\%)$	$\frac{dQ}{d\theta}$	Q(bits-seconds)	$\epsilon_Q(\%)$
29	-990.772	247580908.9	-0.01340	107.1506	14529075	-0.17527
30	-990.905	243189217.8	-0.05206	106.7749	15003135	-0.17498
31	-992.095	238795242.1	-0.00396	106.4005	15475533	-0.17466
32	-992.883	234398102.1		106.0343	15946276	
50	-938.981	156504289	0.44117	98.15068	24117584	-0.30996
51	-932.099	152361085.9	0.41566	97.60793	24551240	-0.34798
52	-924.539	148247196.2	0.35106	96.93297	24982333	-0.36238
53	-917.036	144164022.6		96.17071	25410383	
99	-257.33	21145969.78	2.93241	54.69757	40768507	-1.37272
100	-239.725	20038928.89	2.26058	52.51847	41007599	-1.91512
101	-228.567	19000487.04	1.77388	50.50563	41235903	-0.15232
102	-221.477	18005446.67		50.05078	41459403	

Table 4: IPA sensitivity analysis - the single-stage with competing flow $\theta \equiv b$, $k = 25\%$

	seed=1000					
b (pkts)	$\frac{dL}{d\theta}$	L(bits)	$\epsilon_L(\%)$	$\frac{dQ}{d\theta}$	Q(bits-seconds)	$\epsilon_Q(\%)$
29	-1117.14	203430136	0.15571	102.8726	13836987	-0.31283
30	-1111.55	198486664.8	0.17014	102.1608	14291492	-0.18823
31	-1109.94	193568646.4	0.12992	101.9127	14743417	-0.37695
32	-1105.3	188655800		100.9653	15193391	
50	-896.683	107122797.4	1.19012	86.86244	22770572	-1.14444
51	-872.896	103195995.3	1.45219	85.31209	23151140	-0.58269
52	-847.427	99383499.27	1.23817	84.06976	23527040	-0.90484
53	-827.291	95674206.14		83.02451	23896266	
99	-129.143	9893687.594	3.14784	37.01836	36517926	-1.25435
100	-119.95	9339344.063	4.35321	35.5833	36679933	-1.60033
101	-111.021	8830869.194	5.05585	34.84498	36835115	-4.09009
102	-102.334	8363702.364		32.74303	36983231	
	seed=5586					
b (pkts)	$\frac{dL}{d\theta}$	L(bits)	$\epsilon_L(\%)$	$\frac{dQ}{d\theta}$	Q(bits-seconds)	$\epsilon_Q(\%)$
29	-1123.45	199419510.5	-0.04338	102.7013	13818434	-0.13514
30	-1124.46	194438223.7	0.30869	102.3292	14272991	-0.33593
31	-1119.27	189470011.6	0.16717	101.7407	14724991	-0.24777
32	-1114.21	184517704.6		101.2489	15174788	
50	-867.999	104124585.1	0.81032	83.34979	22577697	-0.61359
51	-851.76	100308787.4	1.01285	82.19303	22944836	-0.63483
52	-829.992	96572023.01	1.08508	81.06608	23306803	-0.89899
53	-813.402	92933411.59		79.88014	23662858	
99	-109.556	8643316.557	3.57180	31.54269	35440102	-2.10866
100	-101.334	8175107.58	0.23725	29.59988	35576951	-3.96068
101	-96.6609	7727062.195	3.38734	27.60289	35702942	-0.66624
102	-91.2591	7313172.6		27.10854	35824463	
	seed=9736					
b (pkts)	$\frac{dL}{d\theta}$	L(bits)	$\epsilon_L(\%)$	$\frac{dQ}{d\theta}$	Q(bits-seconds)	$\epsilon_Q(\%)$
29	-1116.22	202726605.9	0.17726	102.7421	13862567	-0.19978
30	-1109.77	197788294.2	0.17746	102.212	14317010	-0.27933
31	-1103.94	192878520.1	0.33630	101.5427	14768748	-0.27158
32	-1097.75	188002330.7		101.0091	15217563	
50	-884.72	106886080.1	1.13538	86.52321	22781746	-0.73632
51	-864.415	103009518.3	0.43136	85.35367	23162393	-0.84608
52	-850.094	99194954.74	0.78461	84.76994	23537480	-1.14243
53	-835.819	95456900.17		83.69548	23908888	
99	-132.319	10184214.55	2.92398	37.05785	36411614	-4.72862
100	-125.233	9614925.631	-1.87958	35.74658	36568089	-2.64972
101	-119.433	9049458.902	3.10152	34.73654	36722319	-1.29648
102	-112.584	8536549.529		33.09287	36874276	

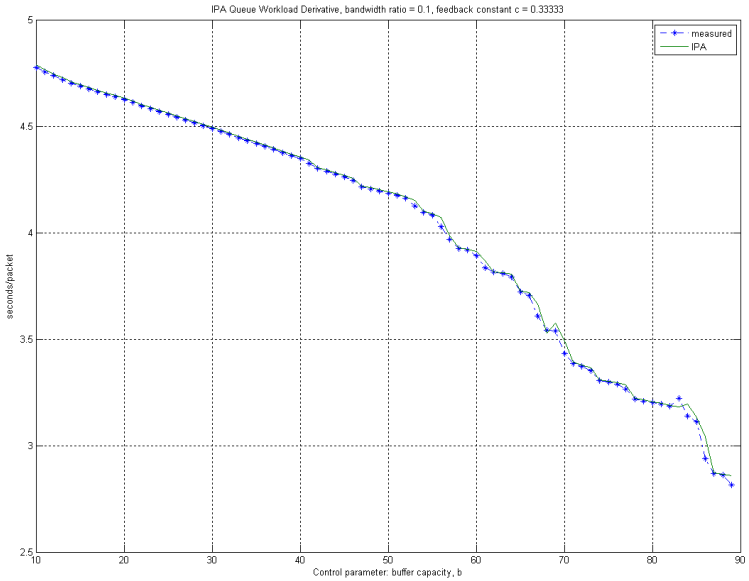


(a) IPA loss-derivative error

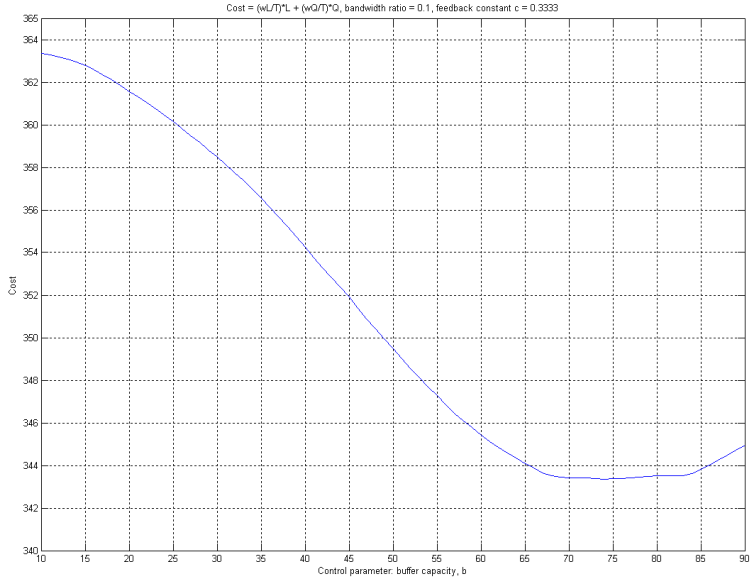


(b) IPA queue-workload-derivative error

Figure 11: IPA derivative results for the single-stage with non-responsive competing flow ($k = 10\%, c = 0.3333$)

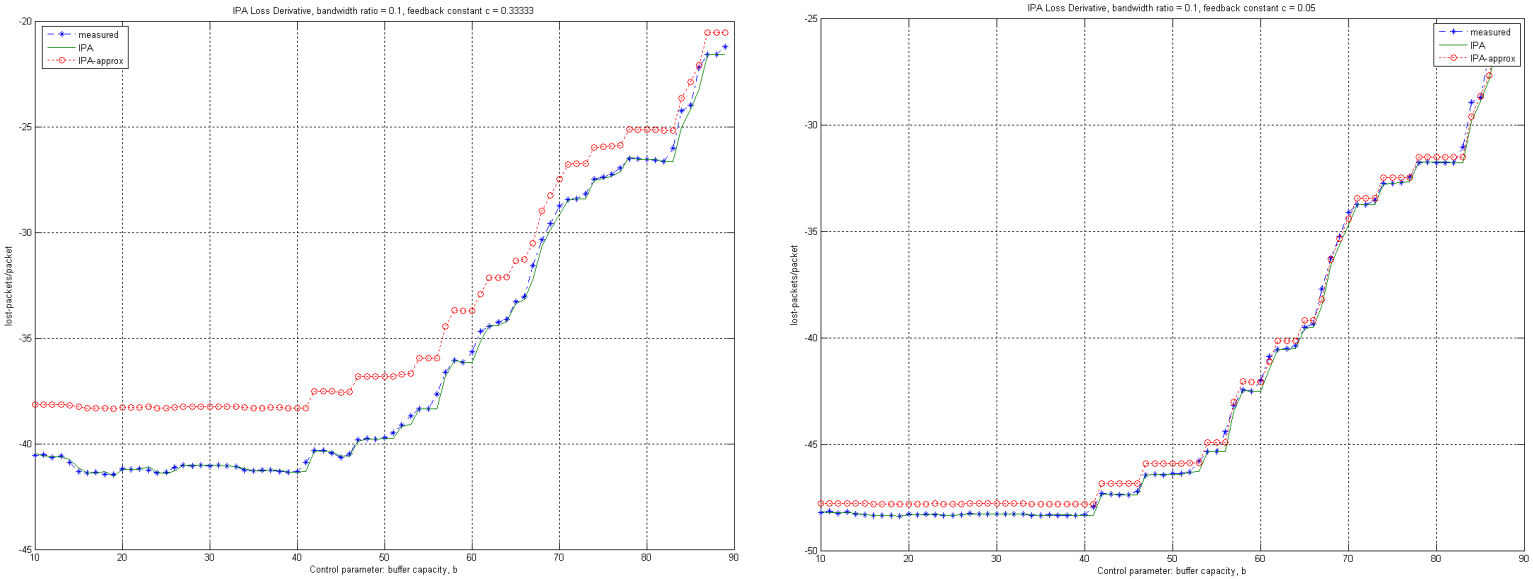


(c) IPA queue-workload-derivative



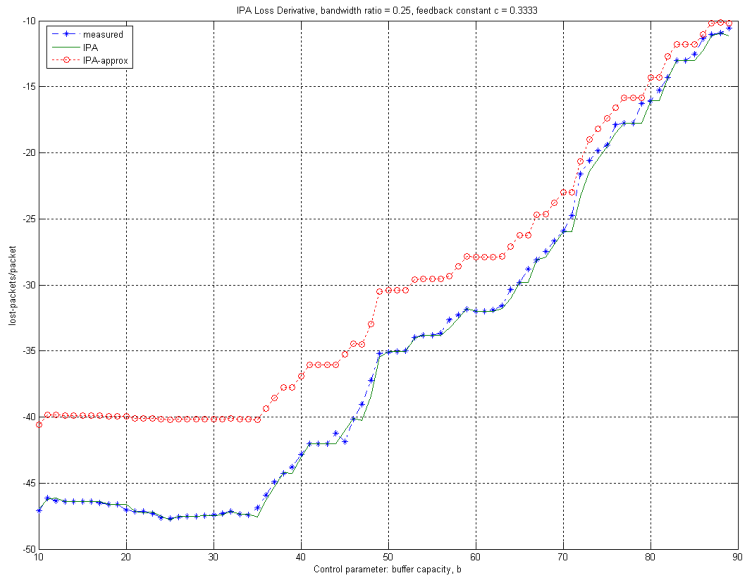
(d) Cost function

Figure 11: IPA derivative results for the single-stage with non-responsive competing flow ($k = 10\%, c = 0.3333$)

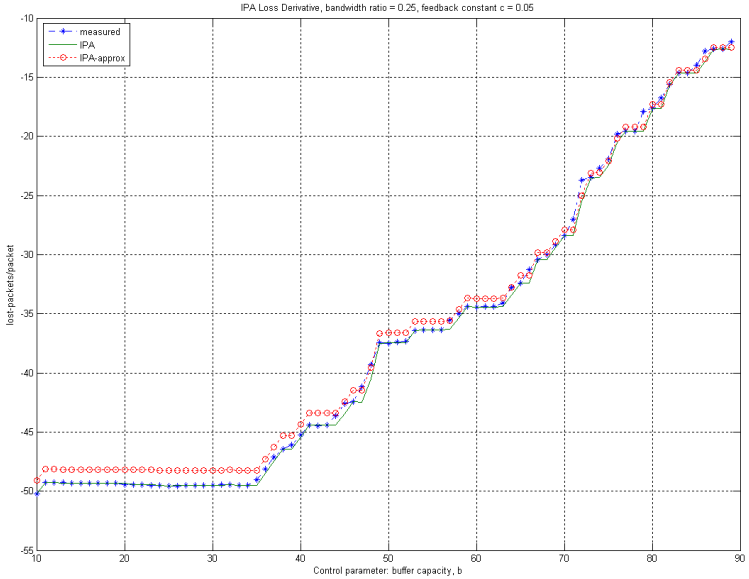


(a) $k = 10\%, c = 0.3333$ (b) $k = 10\%, c = 0.05$

Figure 12: Loss derivative (measured, IPA, IPA-approximated) for different combinations of k and c)



(c) $k = 25\%, c = 0.3333$

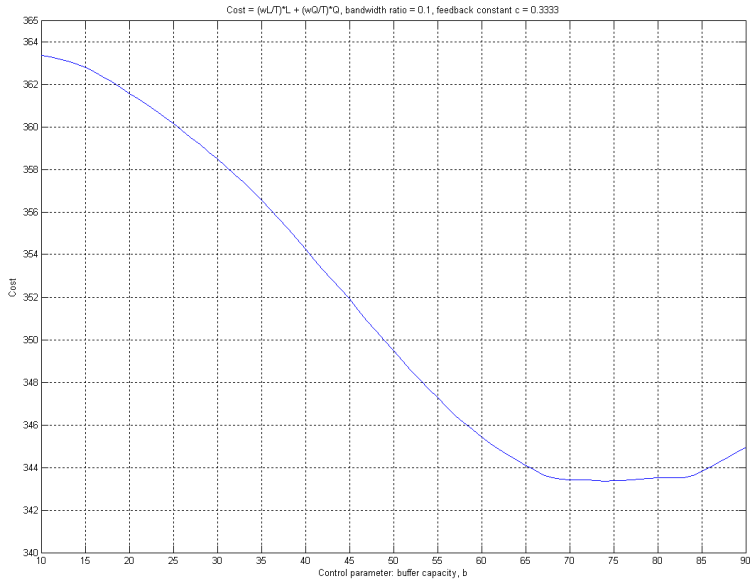


(d) $k = 25\%, c = 0.05$

Figure 12: Loss derivative (measured, IPA, IPA-approximated) for different combinations of k and c)

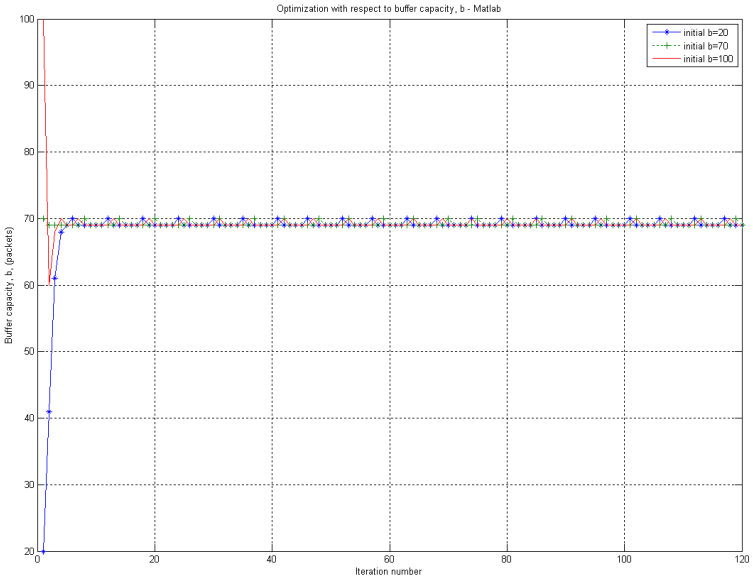
- if $\theta_{i+1} < 2, \theta_{i+1} = 2$.
- if $\theta_{i+1} > 2000, \theta_{i+1} = 2000$

Two cases were considered. For each run, the same seed was used for the random-number generator, therefore the the optimization should converge to the minimum of the cost function as in Figure 13a which was at $b \approx 75$ packets. The next case, has each run completely independent of each other (i.e. a random seed). The results for the first case are seen in Figure 13b. It converges to $b \approx 75$packets. Figure 13c shows the results of the second case. There is still convergence in the presence of noise, but slightly away from the true minimum.

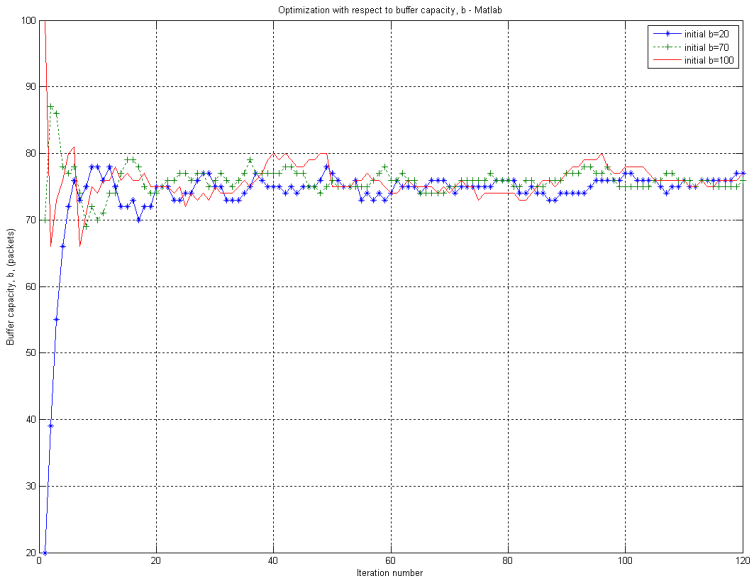


(a) Cost function

Figure 13: IPA optimization with respect to buffer capacity, b

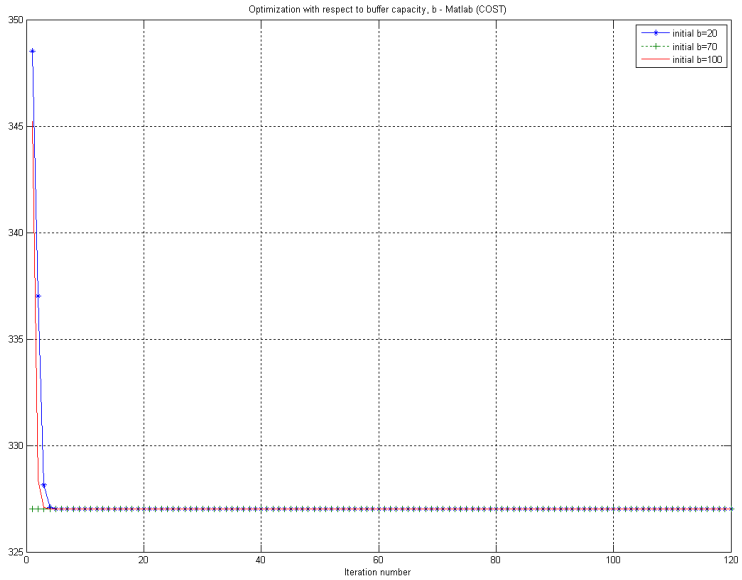


(b) Optimization with same seed

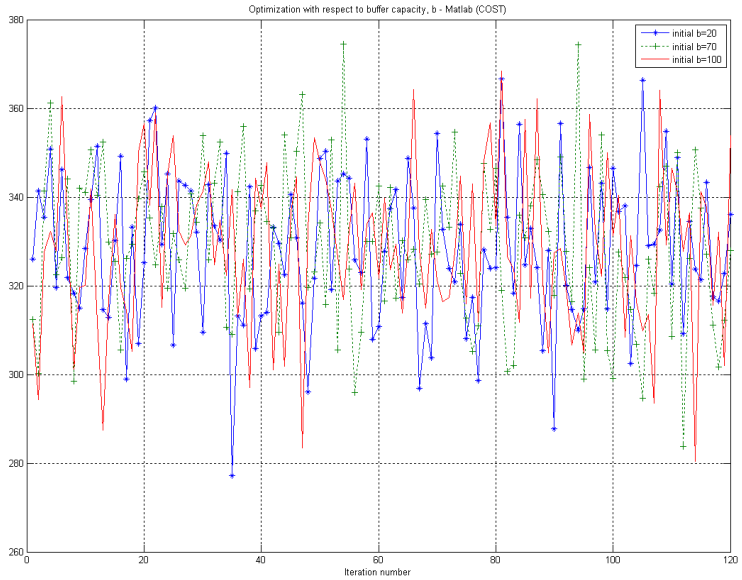


(c) Optimization with randomness

Figure 13: IPA optimization with respect to buffer capacity, b



(d) Cost progression with same seed



(e) Cost progression with randomness

Figure 13: IPA optimization with respect to buffer capacity, b

CHAPTER IV

SINGLE-STAGE FLUID QUEUE WITH DELAYED LOSS FEEDBACK

In this chapter we analyze the single-stage fluid queue with additive *delayed* loss feedback. Using the same notation as in previous chapters, we have the following stochastic time-varying processes: the source rate, $\sigma(t)$, the service rate, $\beta(t)$, the inflow rate, $\alpha(\theta;t)$ (where θ is the control parameter for the optimization), the loss rate, $\gamma(\theta;t)$, the outflow rate, $\delta(\theta;t)$, and the buffer occupancy, $x(\theta;t)$. Again, b and c denote the buffer capacity and the loss-feedback constant respectively. T is the IPA time-horizon such that $T > 0$ and $t \in [0,T]$. We now include an additional parameter, T_d, which is the delay in the feedback. Figure 14 illustrates such an SFM with delay in the loss feedback.

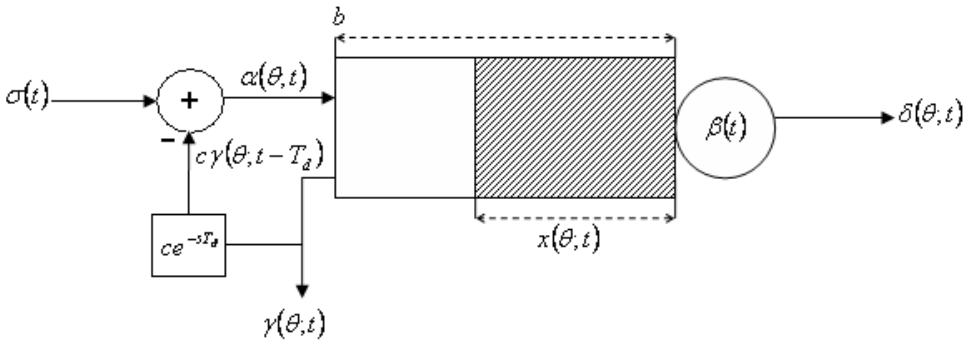


Figure 14: Stochastic fluid model with delayed additive-loss feedback for the single stage case

The dynamics of this system are oulined as follows:

$$\frac{dx}{dt} = \begin{cases} 0 & \text{if } x(\theta;t) = 0 \text{ and } A(\theta;t) < 0 \\ 0 & \text{if } x(\theta;t) = b \text{ and } A(\theta;t) > 0 \\ A(\theta;t) & \text{otherwise} \end{cases} \tag{67}$$

where $A(\theta;t)$ has the same definition as in previous chapters (see Eq(5)). Also:

$$\gamma(\theta;t) = \begin{cases} A(\theta;t) & \text{if } x(\theta;t) = b \\ 0 & \text{otherwise} \end{cases} \tag{68}$$

However, the expression for the inflow rate is different:

$$\alpha(\theta;t) = \sigma(t) - c\gamma(\theta;t - T_d) \tag{69}$$

Before we compute the IPA derivatives for loss volume and queue workload, we return to the definition of different types of events. As before, we define an exogenous event as discontinuities in the defining processes $\sigma(t)$, $\beta(t)$ and endogenous events as when the queue becomes full or becomes empty.

We must now define a new event type, which we will call the induced event. This event causes a jump in the inflow-rate and is due to the loss process some T_d seconds earlier. To elaborate, let z be the time of such an induced event in that $\alpha(\theta;t)$ increases or decreases by $c\gamma(\theta;t - T_d)$. For example, when the queue became full some T_d seconds earlier, $\alpha(\theta;t)$ will switch from $\sigma(t)$ to $\sigma(t) - c\gamma(\theta;t - T_d)$ at $t = z$. Similarly, when the queue ceased to be full at $t = z - T_d$, $\alpha(\theta;t)$ will switch from $\sigma(t) - c\gamma(\theta;t - T_d)$ to $\sigma(t)$ at $t = z$. We have already established that exogenous event-times are independent of θ (where the control parameter, θ is either the buffer capacity, b, or the loss-feedback constant, c). Also, it was said that endogenous event-times are generally dependent on θ. Now, we need to determine the θ-dependence of induced events.

Consider the case when $\alpha(\theta;t)$ switches from $\sigma(t)$ to $\sigma(t) - c\gamma(\theta;t - T_d)$. This would have been due to the queue some time T_d seconds earlier becoming full, and this, in turn, is an endogenous event and hence dependent on θ. Additionally, if this change in $\alpha(\theta;t)$ occurred during a full period, this will cause an θ-dependent change in the loss-rate $\gamma(\theta;t)$ at that time, which will manifest itself a next T_d seconds later as an induced event. This can occur for some kT_d delay times if the full period persists, where $k = 1, 2, \ldots$.

On the other hand, consider the case when the opposite is true, i.e., $\alpha(\theta;t)$ switches from $\sigma(t) - c\gamma(\theta;t - T_d)$ to $\sigma(t)$. This was induced by the queue ceasing to be full some time T_d seconds earlier. With regard to the queue ceasing to be full, we must consider three cases.

For the first case, there was at $t - T_d$ an exogenous event such that there was a negative jump in the inflow rate $A(\theta; t - T_d)$, as a result the queue ceased to be full at that time $t - T_d$. Therefore, in this case, the time for the induced event is independent of θ. For the second case, the queue ceases to be full at $t - T_d$ when $A(\theta; t - T_d)$ is continuous at $t - T_d$ and $A(\theta; t - T_d) = 0$. This event time may or may not be independent of θ, however the term $A(\theta; t - T_d)\frac{d(t-T_d)}{d\theta} = 0$. In the third case, the inducing event at $t - T_d$ (which was the queue ceasing to be full) may have itself been induced by the endogenous event, "queue becomes full", which occurred T_d seconds earlier or by another induced event. The endogenous event, "queue becomes full", will cause $A(\theta; t - T_d)$ to drop by $c\gamma(\theta; t - kT_d)$ where the constant k is loosely used here to represent the number of inducing events that led to the induced event at $t - T_d$.

If we use the same nomenclature of previous chapters and Figure 2 as the example trajectory, we can again express the total loss volume as:

$$L_T(\theta) = \sum_{i=1}^{K} \int_{\tau_i(\theta)}^{\varsigma_i(\theta)} \gamma(\theta; t)\, dt \tag{70}$$

where there are K full-periods in the interval $[0, T]$.

Also, the queue workload can be expressed as:

$$Q(\theta; T) \quad = \quad \int_0^T x(\theta; t)\, dt = \sum_{i=1}^{M} \left[\int_{\varsigma_{i-1}}^{\tau_i} x(\theta; t)\, dt + \int_{\tau_i}^{\varsigma_i} x(\theta; t)\, dt \right] \tag{71}$$

where, the period $[0, T]$ was partitioned into non-boundary periods (NBPs) and boundary periods (BPs) of which there were M BPs.

We use the same assumptions as listed in Chapter 2, for $\frac{dL_T(\theta)}{d\theta}$ to exist. Differentiating Eq. (70) and substituting into it Eq. (68), we have

$$\frac{dL_T(\theta)}{d\theta} = \sum_{i=1}^{K} \frac{d}{d\theta} \left[\int_{\tau_i(\theta)}^{\varsigma_i(\theta)} A(\theta; t)\, dt \right] \tag{72}$$

Additionally, let $z_{i,j}$ be the j-th induced event that is dependent on θ out of J_i such events in the full period $(\tau_i(\theta), \varsigma_i(\theta))$ and let $s_{i,n}$ be the n-th induced event (dependent on θ) out of N_{i-1} such events occuring in the NBP $(\varsigma_{i-1}(\theta), \tau_i(\theta))$ just before the boundary period. Therefore Eq. (72) becomes:

$$\begin{aligned}\frac{dL_T(\theta)}{d\theta} &= \sum_{i=1}^{K}\Bigg[\int_{\tau_i}^{z_{i,1}}\frac{dA(\theta;t)}{d\theta}\,dt + A(\theta;z_{i,1}^{-})\frac{dz_{i,1}}{d\theta} - A(\theta;\tau_i^{+})\frac{d\tau_i}{d\theta} \\ &\quad + \sum_{j=1}^{J_i}\left[\int_{z_{i,j}}^{z_{i,j+1}}\frac{dA(\theta;t)}{d\theta}\,dt + A(\theta;z_{i,j+1}^{-})\frac{dz_{i,j+1}}{d\theta} - A(\theta;z_{i,j}^{+})\frac{dz_{i,j}}{d\theta}\right] \\ &\quad + \int_{z_{i,J_i}}^{\varsigma_i}\frac{dA(\theta;t)}{d\theta}\,dt + A(\theta;\varsigma_i^{-})\frac{d\varsigma_i}{d\theta} - A(\theta;z_{i,J_i}^{+})\frac{dz_{i,J_i}}{d\theta}\Bigg] \qquad (73)\end{aligned}$$

4.1 Buffer capacity, b, as the control parameter, θ

Between queue event-times that are dependent on θ, $A(\theta;t)$ is independent of θ. This is because between these queue event-times, $A(\theta;t)$ is solely a function of the defining processes which are themselves indpendent of θ. As a result, $\frac{dA(\theta;t}{d\theta} = 0$, and all the integral terms disappear in Eq. (73), so that Eq. (73) simplifies to

$$\frac{dL_T(\theta)}{d\theta} = \sum_{i=1}^{K}\left[-A(\theta;\tau_i^{+})\frac{d\tau_i}{d\theta} + A(\theta;\varsigma_i^{-})\frac{d\varsigma_i}{d\theta} + \sum_{j=1}^{J_i}\left(A(\theta;z_{i,j}^{-}) - A(\theta;z_{i,j}^{+})\right)\frac{dz_{i,j}}{d\theta}\right] \qquad (74)$$

If the discontinuity in $A(\theta;t)$ at $t = z_{i,j}$ is due to the queue becoming full kT_d ($k = 1, 2, \ldots$) seconds earlier, then

$$A(\theta;z_{i,j}^{-}) = \sigma(z_{i,j}) - c\gamma(\theta;(z_{i,j} - T_d)^{-}) - \beta(z_{i,j}) \qquad (75)$$

$$A(\theta;z_{i,j}^{+}) = \sigma(z_{i,j}) - c\gamma(\theta;(z_{i,j} - T_d)^{+}) - \beta(z_{i,j}) \qquad (76)$$

so that

$$A(\theta;z_{i,j}^{-}) - A(\theta;z_{i,j}^{+}) = -c(\gamma(\theta;(z_{i,j} - T_d)^{-}) - \gamma(\theta;(z_{i,j} - T_d)^{+})) \qquad (77)$$

Now

$$\gamma(\theta;(z_{i,j} - T_d)^{-}) = \begin{cases} 0 & \text{if at } t = z_{i,j} - T_d \text{ the queue became full} \\ A(\theta;(z_{i,j} - T_d)^{-}) & \text{if at } t = z_{i,j} - T_d \text{ the queue is in a full period} \end{cases} \qquad (78)$$

$$\gamma(\theta;(z_{i,j} - T_d)^{+}) = A(\theta;(z_{i,j} - T_d)^{+}) \qquad (79)$$

If $\gamma(\theta;(z_{i,j}-T_d)^-)$ is non-zero, then consider the time instant T_d seconds earlier, i.e., $t = z_{i,j} - 2T_d$. Therefore

$$A(\theta;(z_{i,j}-T_d)^-) - A(\theta;(z_{i,j}-T_d)^+) \quad = \quad -c(\gamma(\theta;(z_{i,j}-2T_d)^-) - \gamma(\theta;(z_{i,j}-2T_d)^+)) \tag{80}$$

where

$$\gamma(\theta;(z_{i,j}-2T_d)^-) \quad = \quad \begin{cases} 0 & \text{if at } t = z_{i,j}-2T_d \text{ the queue became full} \\ A(\theta;(z_{i,j}-2T_d)^-) & \text{if at } t = z_{i,j}-2T_d \text{ the queue is in a full period} \end{cases} \tag{81}$$

$$\gamma(\theta;(z_{i,j}-2T_d)^+) \quad = \quad A(\theta;(z_{i,j}-2T_d)^+) \tag{82}$$

Continuing recursively in this fashion, it can be shown that:

$$A(\theta;z_{i,j}^-) - A(\theta;z_{i,j}^+) \quad = \quad (-c)^{k-1}cA(\theta;(z_{i,j}-kT_d)) \tag{83}$$

where at $t = z_{i,j} - kT_d = \tau_i$ is the time the queue became full.

Also $\frac{d}{d\theta}(z_{i,j}-kT_d) = \frac{dz_{i,j}}{d\theta}$, assuming that T_d is independent of θ.

If the discontinuity in $A(\theta;t)$ at $t = z_{i,j}$ is due to the queue ceasing to be full some T_d seconds earlier, then

$$A(\theta;z_{i,j}^-) - A(\theta;z_{i,j}^+) \quad = \quad -cA(\theta;z_{i,j}-T_d) \tag{84}$$

Now, consider the NBP just before the full period $(\tau_i(\theta),\varsigma_i(\theta))$, i.e., $(\varsigma_{i-1}(\theta),\tau_i(\theta))$:

$$x(\theta;t)|_{\varsigma_{i-1}(\theta)}^{\tau_i(\theta)} \quad = \quad \begin{cases} \theta & \text{if } x(\theta;\tau_i(\theta)) = b \text{ and } x(\theta;\varsigma_{i-1}(\theta)) = 0 \\ 0 & \text{if } x(\theta;\tau_i(\theta)) = b \text{ and } x(\theta;\varsigma_{i-1}(\theta)) = b \end{cases} \tag{85}$$

Differentiationg with respect to θ:

$$\frac{dx(\theta;t)}{d\theta} \quad = \quad \begin{cases} 1 & \text{if } x(\theta;\tau_i(\theta)) = b \text{ and } x(\theta;\varsigma_{i-1}(\theta)) = 0 \\ 0 & \text{if } x(\theta;\tau_i(\theta)) = b \text{ and } x(\theta;\varsigma_{i-1}(\theta)) = b \end{cases} \tag{86}$$

Also, from the system dynamics in Eq. (67), it can be seen that for the NBP, $\frac{dx}{dt} = A(\theta;t)$

$$\begin{aligned} \frac{dx}{d\theta} &= \frac{d}{d\theta}\left(\int_{\varsigma_{i-1}(\theta)}^{\tau_i(\theta)} \frac{dx(\theta,t)}{dt}\,dt\right) \\ &= \frac{d}{d\theta}\left(\int_{\varsigma_{i-1}(\theta)}^{\tau_i(\theta)} A(\theta;t)\,dt\right) \end{aligned} \tag{87}$$

Proceeding as we did for the analysis of the full period:

$$\frac{dx}{d\theta} = -A(\theta;\varsigma_{i-1}^{+})\frac{d\varsigma_{i-1}}{d\theta} + A(\theta;\tau_i^{-})\frac{d\tau_i}{d\theta} + \sum_{j=1}^{N_{i-1}}\left(A(\theta;s_{i,j}^{-}) - A(\theta;s_{i,j}^{+})\right)\frac{ds_{i,j}}{d\theta} \quad (88)$$

This leads to:

$$A(\theta;\tau_i^{-})\frac{d\tau_i}{d\theta} = \pi_i + A(\theta;\varsigma_{i-1}^{+})\frac{d\varsigma_{i-1}}{d\theta} - \sum_{j=1}^{N_{i-1}}\left(A(\theta;s_{i,j}^{-}) - A(\theta;s_{i,j}^{+})\right)\frac{ds_{i,j}}{d\theta} \quad (89)$$

where

$$\pi_i = \begin{cases} 1 & \text{if } x(\theta;\tau_i(\theta)) = b \text{ and } x(\theta;\varsigma_{i-1}(\theta)) = 0 \\ 0 & \text{if } x(\theta;\tau_i(\theta)) = b \text{ and } x(\theta;\varsigma_{i-1}(\theta)) = b \end{cases} \quad (90)$$

At $t = \tau_i(\theta)$, only an endogenous event can occur (which in this case, the queue becomes full), as a result, $A(\theta;t)$ is continuous at $t = \tau_i(\theta)$ so that $A(\theta;\tau_i^{-}) = A(\theta;\tau_i^{+})$. Consequently, Eq. (89) can be substituted into Eq. (74) with the ensuing expression being:

$$\frac{dL_T(\theta)}{d\theta} = \sum_{i=1}^{K}\left[-\pi_i - A(\theta;\varsigma_{i-1}^{+})\frac{d\varsigma_{i-1}}{d\theta} + \sum_{j=1}^{N_{i-1}}\left(A(\theta;s_{i,j}^{-}) - A(\theta;s_{i,j}^{+})\right)\frac{ds_{i,j}}{d\theta} + A(\theta;\varsigma_i^{-})\frac{d\varsigma_i}{d\theta} + \sum_{j=1}^{J_i}\left(A(\theta;z_{i,j}^{-}) - A(\theta;z_{i,j}^{+})\right)\frac{dz_{i,j}}{d\theta}\right] \quad (91)$$

$\left(A(\theta;s_{i,j}^{-}) - A(\theta;s_{i,j}^{+})\right)\frac{ds_{i,j}}{d\theta}$ follows the same relationship as in Eq. (83).

It should be noted that:

$$A(\theta;\varsigma_{i-1}^{+})\frac{d\varsigma_{i-1}}{d\theta} = \begin{cases} 0 & \text{if not an induced event} \\ \rho_{i-1}^{+} & \text{if it is an induced event} \end{cases} \quad (92)$$

$$A(\theta;\varsigma_i^{-})\frac{d\varsigma_i}{d\theta} = \begin{cases} 0 & \text{if not an induced event} \\ \rho_i^{-} & \text{if it is an induced event} \end{cases} \quad (93)$$

where

$$\rho_i^{-} = \frac{A(\theta;\varsigma_i^{-})}{A(\theta;(\varsigma_i - T_d)^{+})}\left(A(\theta;(\varsigma_i - T_d)^{+})\frac{d(\varsigma_i - T_d)}{d\theta}\right) \quad (94)$$

$$\rho_{i-1}^{+} = \frac{A(\theta;\varsigma_{i-1}^{+})}{A(\theta;(\varsigma_{i-1} - T_d)^{+})}\left(A(\theta;(\varsigma_{i-1} - T_d)^{+})\frac{d(\varsigma_{i-1} - T_d)}{d\theta}\right) \quad (95)$$

The analysis for the queue workload is carried it out in detail in Chapter 5 and incompasses the concept of induced events. Therefore, we will only present here the result for the queue workload for the single-stage delay case:

$$\begin{aligned}\frac{dQ(\theta;T)}{d\theta} &= \sum_{i=1}^{M}\Bigg[\left(\frac{dx(\theta;\varsigma_{i-1})}{d\theta} - A(\theta;\varsigma_{i-1}^{+})\frac{d\varsigma_{i-1}}{d\theta}\right)(\tau_i - \varsigma_{i-1}) \\ &\quad + \sum_{j=1}^{N_{i-1}}\left((A(\theta;s_{i,j}^{-}) - A(\theta;s_{i,j}^{+}))\frac{ds_{i,j}}{d\theta}(\tau_i - s_{i,j})\right) \\ &\quad + \frac{dx(\theta;\tau_i)}{d\theta}(\varsigma_i - \tau_i)\Bigg] \end{aligned} \tag{96}$$

where

$$\frac{dx(\theta;\tau_i)}{d\theta} = \begin{cases} 0 & \text{if } [\tau_i, \varsigma_i] \text{ is an empty period} \\ 1 & \text{if } [\tau_i, \varsigma_i] \text{ is a full period} \end{cases} \tag{97}$$

$$\frac{dx(\theta;\varsigma_{i-1})}{d\theta} = \begin{cases} 0 & \text{if } x(\theta;\varsigma_{i-1}(\theta)) = 0 \\ 1 & \text{if } x(\theta;\varsigma_{i-1}(\theta)) = b \end{cases} \tag{98}$$

4.2 Simulations with buffer capacity, b, as the control variable θ

The test configuration (Figure 3), parameters and procedures were the same as that for the single-stage case without delay in the feedback loop. (See Chapter 2, Section 2.3.) The time-horizon for the optimizations was, $T = 10$ second. The loss-feedback delay, $T_d = 0.01s$. For both the error analysis and optimization Matlab was used.

4.2.1 IPA derivative error-analysis

Simulations were run for $T = 240$ seconds for three sets of four adjacent values of θ (in packets), namely $\{29, 30, 31, 32\}$, $\{50, 51, 52, 53\}$ and $\{99, 100, 101, 102\}$, and the IPA derivatives were computed in each case. The error between the first-order approximation of the derivative and that of the IPA was also calculated. This process was repeated for three different seed values of the random number generator. The error values are presented in Table 5.

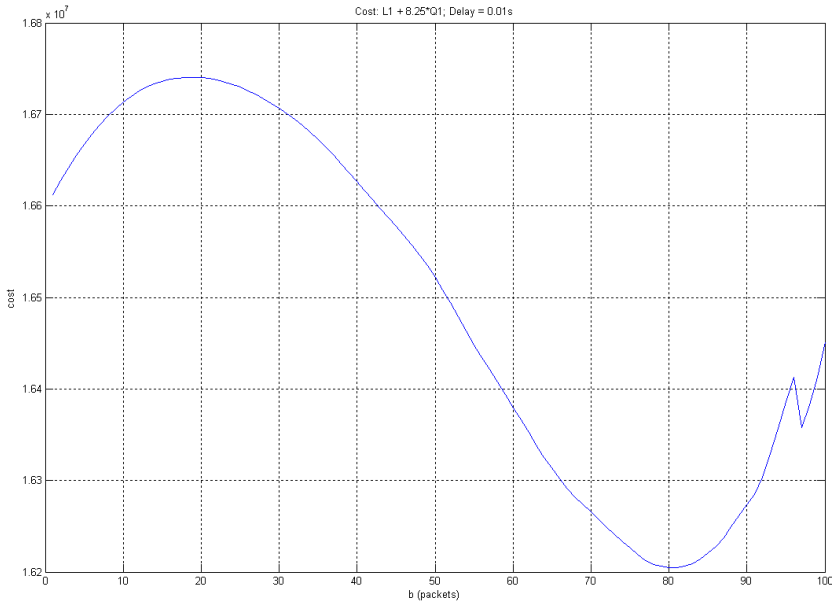
4.2.2 IPA optimization

The IPA optimization followed the same procedure as in Section 2.3.2, however the initial stepsize was $a_0 = 10$ instead of 5. The simulations were run for 120 iterations, with each iteration having a time-horizon of $T = 10$ seconds. The initial values of b were 20, 70

Table 5: IPA sensitivity analysis - the single-stage case with delay $\theta \equiv b, T_d = 0.01$s

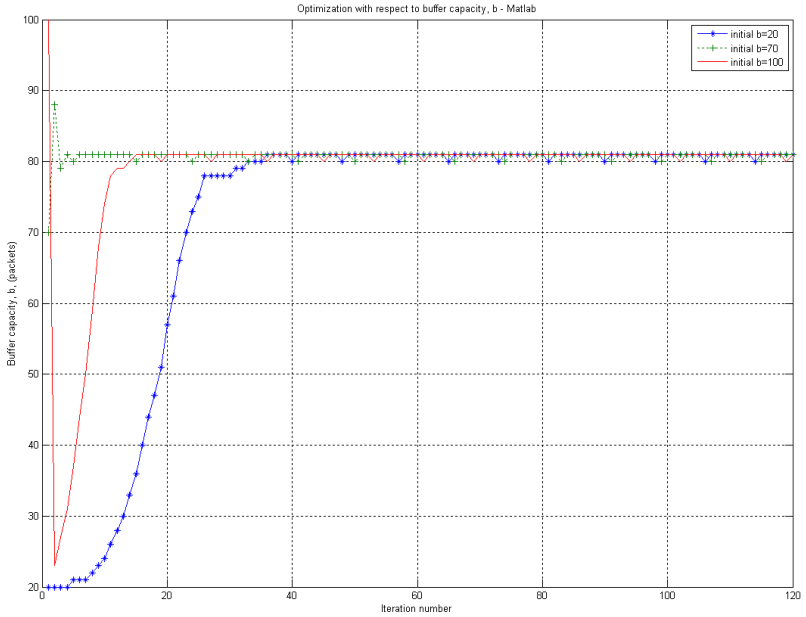
	seed=1000					
b (pkts)	$\frac{dL}{d\theta}$	L(bits)	$\epsilon_L(\%)$	$\frac{dQ}{d\theta}$	Q(bits-seconds)	$\epsilon_Q(\%)$
29	-889.225	282576312	-1.21773	107.4122	14503879	-0.15042
30	-890.473	278587277	-1.08183	107.0882	14979214	-0.15217
31	-889.785	274598003	-1.13829	106.7598	15453107	-0.15385
32	-890.422	270609587		106.4308	15925538	
50	-890.532	198817665	-1.10153	100.9506	24188086	-0.12892
51	-889.12	194827352	-1.07103	100.6601	24634922	-0.20161
52	-888.891	190844566	-1.00433	100.2144	25080148	-0.08899
53	-888.414	186865433		99.97635	25523903	
99	-402.751	40172711	2.17300	64.72377	42765585	-1.18534
100	-387.886	38426508	1.51075	63.36174	43049041	0.31932
101	-376.247	36733367	0.52428	63.71013	43330757	0.36292
102	-369.591	35074583		63.87091	43614145	
	seed=5586					
b (pkts)	$\frac{dL}{d\theta}$	L(bits)	$\epsilon_L(\%)$	$\frac{dQ}{d\theta}$	Q(bits-seconds)	$\epsilon_Q(\%)$
29	-890.917	279055146	-1.01175	106.9641	14475898	-0.15898
30	-889.257	275066651	-1.21827	106.6226	14949210	-0.16260
31	-890.391	271077451	-1.06991	106.2781	15420992	-0.16324
32	-890.047	267089017		105.9311	15891248	
50	-886.133	195349345	-1.21810	99.9426	24097888	-0.12301
51	-887.964	191374166	-0.98035	99.6662	24540288	-0.17210
52	-888.295	187400128	-0.97255	99.31474	24981249	-0.08893
53	-889.972	183424918		99.13628	25421020	
99	-421.065	38583962	-1.93555	62.87418	42203937	-2.71168
100	-402.368	36681680	1.44776	61.23993	42475039	0.01668
101	-387.815	34924203	2.26181	61.08167	42746500	-1.89757
102	-369.073	33244282		59.51294	43012077	
	seed=9736					
b (pkts)	$\frac{dL}{d\theta}$	L(bits)	$\epsilon_L(\%)$	$\frac{dQ}{d\theta}$	Q(bits-seconds)	$\epsilon_Q(\%)$
29	-891.22	282231638	-0.99760	107.4838	14508026	-0.15396
30	-890.207	278242346	-1.11186	107.1539	14983660	-0.15318
31	-893.254	274253082	-0.75548	106.8263	15457839	-0.15245
32	-889.472	270264272		106.4999	15930571	
50	-891.378	198509148	-1.13438	101.0274	24197691	-0.08254
51	-893.658	194513745	-1.07743	100.828	24645075	-0.10007
52	-893.161	190510379	-0.90243	100.5086	25091498	-0.15516
53	-887.957	186516165		100.0656	25536261	
99	-423.083	39834762	1.40816	65.99867	42712660	-0.81876
100	-405.526	37986064	0.84184	65.12915	43002771	-1.32444
101	-392.421	36203904	0.54713	63.68338	43287601	-0.49090
102	-380.471	34474209		63.28584	43568460	

and 100 packets. The cost function was $\frac{1}{T}(w_L L(\theta;T) + w_Q Q(\theta;T))$ where $w_L = 10.0$ and $w_Q = 82.5$.

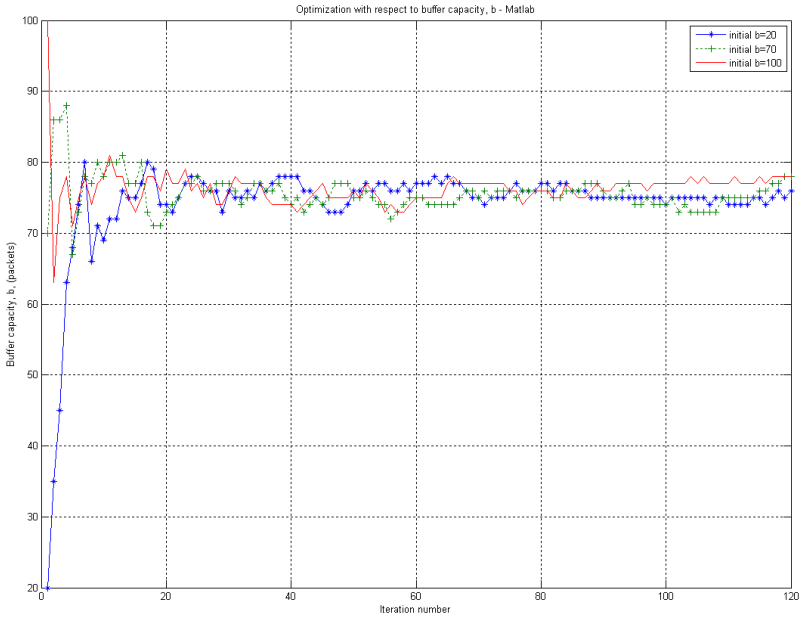


(a) Cost (seed = 1000)

Figure 15: IPA optimization with respect to buffer capacity, b - single-stage case with loss-feedback delay ($T_d = 0.01 second$)

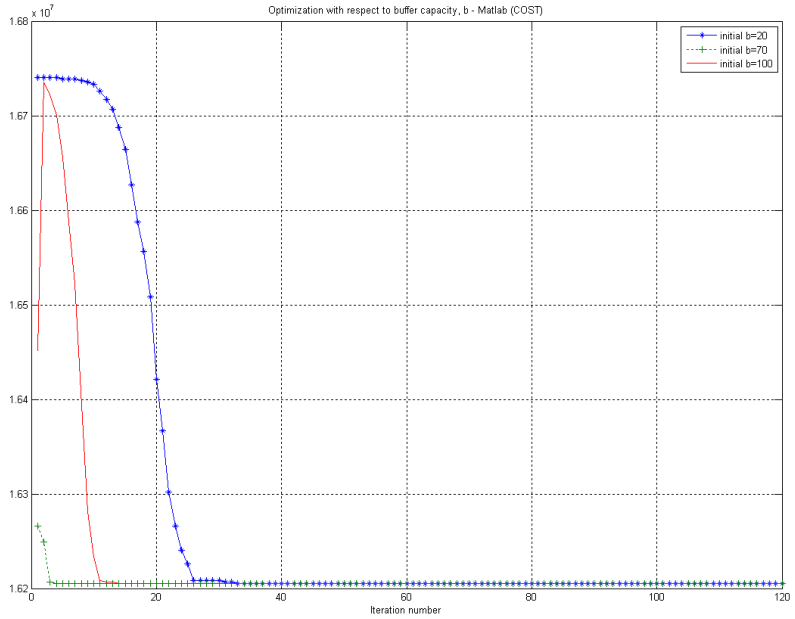


(b) Convergence - deterministic

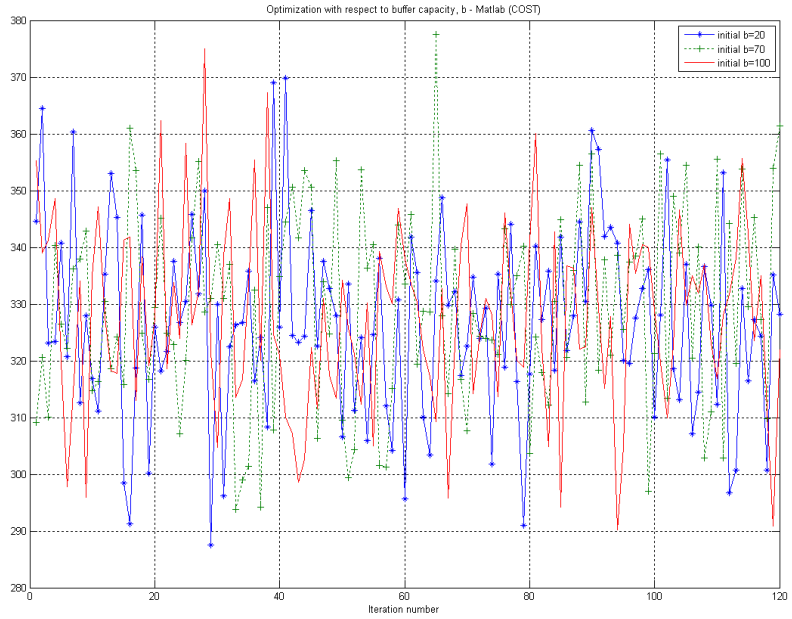


(c) Convergence - random

Figure 15: IPA optimization with respect to buffer capacity, b - single-stage case with loss-feedback delay ($T_d = 0.01 second$)



(d) Cost progression - deterministic



(e) Cost progression - random

Figure 15: IPA optimization with respect to buffer capacity, b - single-stage case with loss-feedback delay ($T_d = 0.01 second$)

CHAPTER V

MULTI-STAGE TANDEM NETWORK OF FLUID QUEUES WITH LOSS FEEDBACK

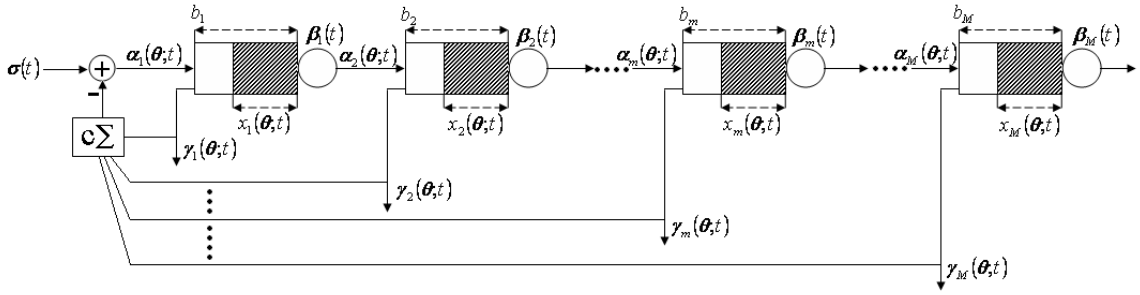


Figure 16: Stochastic fluid model with instantaneous additive-loss feedback for the multi-stage tandem case

5.1 General description

We consider a tandem network of M SFM nodes (queues) indexed by $m = 1, \ldots, M$. (See Figure 16.) Each node has a single input, a single server and a single queue with finite capacity, b_m, where $b_m > 0$.

For $1 < m < M$, the inflow to node m is the outflow from node $m-1$. However, due to the negative additive feedback of instantaneous loss rates (or overflow rates) at each node $\gamma_m(\theta;t), m = 1, \ldots, M$, the instantaneous source rate, $\sigma(t)$, is reduced by the sum of these loss-rates scaled by a constant c called the loss-feedback constant.

The control parameter (which is adjusted for instance for congestion control) is simply denoted by θ where θ can be chosen as $b_m, m = 1, \ldots, M$ or as the loss-feedback constant, c, or even as some parameter of the service rate $\beta_m(t), m = 1, \ldots, M$. As a result, the inflow rate into each queue, $\alpha_m(t), m = 1, \ldots, M$ can in general be a function of θ and hence will be denoted as $\alpha_m(\theta;t)$. Unless some aspect of $\beta_m(t)$ is the control parameter then $\beta_m(t)$ is independent of θ.

The buffer-occupancy process $x_m(\theta;t)$ at node m, the overflow rate, $\gamma_m(\theta;t)$, at node m and

the outflow rate,$\delta_m(\theta;t)$, at node m, are also generally dependent on θ and are therefore denoted as such. The source rate, $\sigma(t)$ is an external process and is independent of θ.

It is assumed that the control parameter θ is real-valued and is confined to a closed and bounded (compact) interval Θ.

For the IPA, we have to analyze the internode effects which occur both in the downstream direction (due to the tandem topology) and in the upstream direction (due to the feedback). For consistency with previous work in IPA-SFM tandem networks we use similar definitions and nomenclature as in [78] for the analysis of this case with loss-feedback.

5.2 Queue dynamics

Consider node m:

- The inflow process $\{\alpha_m(\theta;t)\}$ and service process $\{\beta_m(t)\}$ define the local dynamics at node m and are therefore called the *defining processes* of node m [78].
- Derived from these defining processes are the buffer-occupancy process, $\{x_m(\theta;t)\}$, the outflow process, $\{\delta_m(\theta;t)\}$, and the overflow process, $\{\gamma_m(\theta;t)\}$. These are the *derived processes* at node m [78].
- The stochastic processes $\{\sigma(t)\}$ and $\{\beta_m(t)\}$ are defined over the probability space $(\Omega, \mathcal{F}, \mathcal{P})$. They are random instantaneous rates.

Define:

$$A_m(\theta;t) = \alpha_m(\theta;t) - \beta_m(\theta;t). \text{ where } m = 1, \ldots, M \tag{99}$$

The dynamics of the buffer-occupancy process $\{x_m(\theta;t)\}$, at node m, $m = 1, \ldots, M$ are given by the differential equation:

$$\frac{dx_m}{dt} = \begin{cases} 0 & \text{if } x_m(\theta;t) = 0 \text{ and } A_m(\theta;t) \leq 0 \\ 0 & \text{if } x_m(\theta;t) = b_m \text{ and } A_m(\theta;t) \geq 0 \\ A_m(\theta;t) & \text{otherwise} \end{cases} \tag{100}$$

Now:

$$\alpha_m(\theta;t) = \begin{cases} \delta_{m-1}(\theta;t) & \text{if } m = 2, \ldots, M \\ \max\{\sigma(t) - c\sum_{j=1}^{M} \gamma_j(\theta;t), 0\} & \text{if } m = 1 \end{cases} \tag{101}$$

where

$$\delta_m(\theta;t) = \begin{cases} \beta_{m-1}(t) & \text{if } x_{m-1}(\theta;t) > 0 \\ \alpha_{m-1}(\theta;t) & \text{if } x_{m-1}(\theta;t) = 0 \end{cases} \tag{102}$$

and where

$$\gamma_m(\theta;t) = \begin{cases} A_m(\theta;t) & \text{if } x_m(\theta;t) = b_m \text{ and } A_m(\theta;t) \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{103}$$

These dynamics are tracked over a given finite time-interval (time-horizon) such that $t \in [0,T]$ where $0 < T < \infty$.

To simplify the analysis/discussion we assume that for $m = 1$

$$\alpha_m(\theta;t) = \sigma(t) - c\sum_{j=1}^{M} \gamma_j(\theta;t) \tag{104}$$

in that the congestion control action never forces the inflow rate at the first node to zero.

5.3 Sample performance functions

Two sample performance functions of interest over the time interval $[0,T]$ are

1. The loss volume, $L_m(\theta;T)$ at node m, $m = 1,\ldots,M$ where

$$L_m(\theta;T) = \int_0^T \gamma_m(\theta;t)\,dt \tag{105}$$

2. The queue workload, $Q_m(\theta;T)$ at node m, $m = 1,\ldots,M$ where

$$Q_m(\theta;T) = \int_0^T x_m(\theta;t)\,dt \tag{106}$$

These functions are related to the average loss rate, $\frac{1}{T}E[L]$, and the average buffer occupancy, $\frac{1}{T}E[Q]$, which play a role in characterizing congestion in networking applications. Using IPA we wish to determine the derivatives of the sample performance functions with respect to the control parameter for the purpose of online optimization, i.e. $\frac{dL_m(\theta;T)}{d\theta}$ and $\frac{dQ_m(\theta;T)}{d\theta}$ for $m = 1,\ldots,M$. These IPA sample derivatives are estimators of the derivatives of the performance metrics $E[L_m(\theta;T)]$ and $E[Q_m(\theta;T)]$.

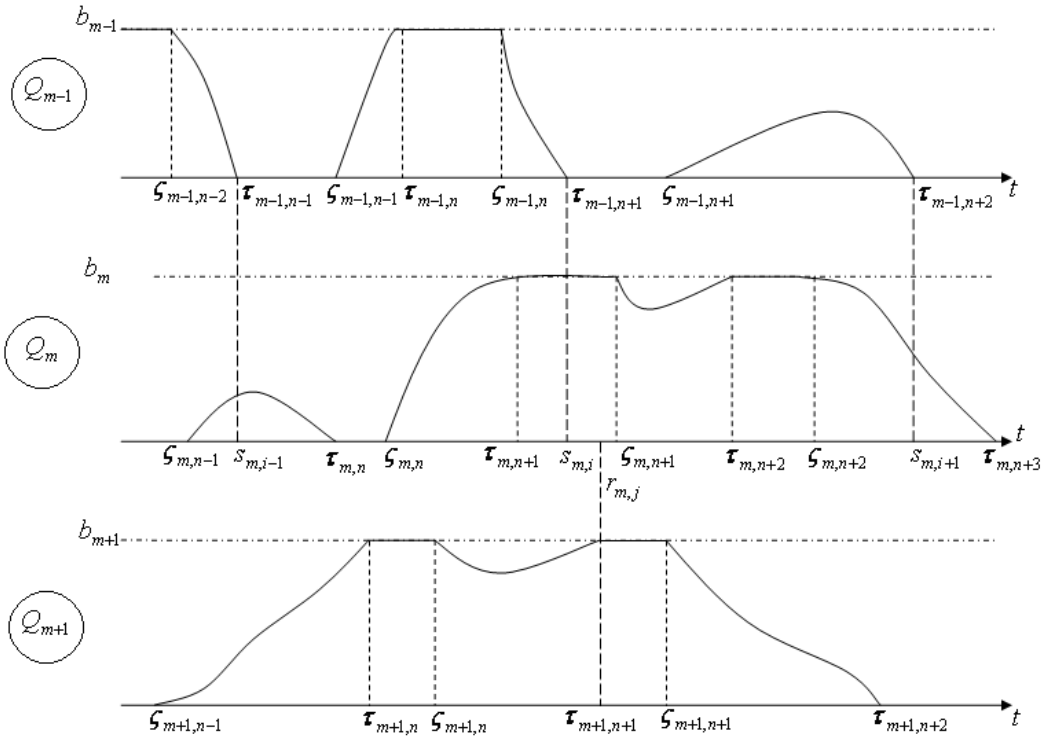


Figure 17: Sample paths of three consecutive queues

5.4 Boundary periods and non-boundary periods

The IPA derivative estimate of the perfomance metric is computed based on an observed sample path (ω) of the buffer occupancy trajectory over time.

Consider an example of such a trajectory at node m in Figure 17. It consists of boundary periods (BPs) alternating with non-boundary periods (NBPs) within the period $[0, T]$. A boundary period can be either an empty period (EP), which is the time-interval during which the queue at node m (Q_m) is empty (i.e. $x_m(\theta; t) = 0$), or a full period (FP), which is the time-interval during which Q_m is full (i.e. $x_m(\theta; t) = b_m$). A non-boundary period is that time-interval during which Q_m is neither full nor empty (i.e. $0 < x_m(\theta; t) < b_m$).

The function $x_m(\theta; t)$ is generally continuous with respect to time, t, for a given value of θ. Therefore BPs are taken to be closed intervals within [0,T] and NBPs to be open intervals. Therefore denote, the n-th BP, $n = 1, \ldots, N_m$ as

$$B_{m,n} = [\tau_{m,n}(\theta), \varsigma_{m,n}(\theta)] \tag{107}$$

where $\tau_{m,n}(\theta)$ is the beginning of the n-th BP in Q_m and $\varsigma_{m,n}(\theta)$ is the end of the same BP, and where N_m is the total number of BPs in [0,T] which is a random number.

Similarly denote the NBP that just precedes the n-th BP as

$$NB_{m,n} = (\varsigma_{m,n-1}(\theta), \tau_{m,n}(\theta)) \tag{108}$$

For convenience let $\varsigma_{m,0}(\theta) = 0$ and $\varsigma_{m,N_m}(\theta) = T$.

For notational economy, we will simply say from now on $\tau_{m,n}$ and $\varsigma_{m,n}$ with the understanding that generally they are dependent on θ.

There are four (4) different types of NBPs depending on the value of $x_m(\theta; t)$ at its beginning and end. Using E to mean "Empty" and F to mean "Full":

1. $EE_{m,n}$: $x_m(\theta; \varsigma_{m,n-1}) = 0$ and $x_m(\theta; \tau_{m,n}) = 0$
2. $EF_{m,n}$: $x_m(\theta; \varsigma_{m,n-1}) = 0$ and $x_m(\theta; \tau_{m,n}) = b_m$
3. $FE_{m,n}$: $x_m(\theta; \varsigma_{m,n-1}) = b_m$ and $x_m(\theta; \tau_{m,n}) = 0$
4. $FF_{m,n}$: $x_m(\theta; \varsigma_{m,n-1}) = b_m$ and $x_m(\theta; \tau_{m,n}) = b_m$

Similar definitions of the BP and NBPs can be found in [78].

5.5 Definition of events

IPA exploits the discrete-event structure of the system. The system dynamics depend on a number of events such as when queues become full or become empty, cease to be full or cease to be empty. In more general terms, the events are when queues switch over from boundary periods to non-boundary periods and vice versa. Discontinuities in the source rates and service rates at the queues are also considered to be events,

We define three types of events that can occur at node m, $m = 1, \ldots, M$, and which account for the all the possible discontinuities of $A_m(\theta; t)$ at Q_m.

1. e_1 (exogenous event): a discontinuity (or jump) in the external processes, $\{\sigma(t)\}$ and $\{\beta_m(t)\}, m = 1, \ldots, M$.
2. e_2 (endogenous event): the beginning or end of boundary periods at Q_m. Here we have two sub-types:

- e_{2-I} (type-1 endogenous event): the beginning of a boundary period (i.e. when Q_m becomes full or becomes empty).
- e_{2-II} (type-2 endogenous event): the end of a boundary period (i.e. when Q_m ceases to be full or ceases to be empty). This particular event can occur in either of two ways:
  (a) Due to a discontinuity (jump) in $\{\sigma(t)\}$ and $\{\beta_i(t)\}, i = 1, \ldots, M$ such that there is a jump in $A_m(\theta; t)$ so much so that its sign changes; or
  (b) At the time-instant when $A_m(\theta; t)$ becomes zero with no discontinuity in $A_m(\theta; t)$ at t.

3. e_3 (induced event): the end of a boundary period at Q_m induced by $Q_i, i = 1, \ldots, M, i \neq m$ when the latter at that time is at the beginning or end of an FP or EP.

5.5.1 More on induced events

To further explain the induced event at Q_m, consider two situations:

1. When $Q_j, j = 1, \ldots, M, j < m$ becomes empty or ceases to be empty and all queues between Q_j and Q_m, i.e. $Q_i, i = j + 1, \ldots, m - 1$ were empty at the time.

2. When $Q_k, j = 2, \ldots, M, k > m$ becomes full or ceases to be full and all queues in front of Q_m, i.e. $Q_l, l = 1, \ldots, m - 1$ were empty at the time.

In the first case, the output rate of Q_j (which will be the input rate of Q_m, $\alpha_m(\theta; t)$) will switch from $\beta_j(t)$ to $\alpha_j(\theta; t)$ where $\alpha_j(\theta; t) < \beta_j(t)$. This drop in $\alpha_m(\theta; t)$ can be significant enough to cause $A_m(\theta; t)$ to switch sign (i.e. from positive to negative), so that if Q_m was full it suddenly ceases to be full. This resulting event in Q_m was said to be induced by that in Q_j.

Similarly, if Q_j ceased to be empty so that $\alpha_m(\theta; t)$ jumps in the positive direction from $\alpha_j(\theta; t)$ to $\beta_j(t)$, then if Q_m was empty at the time, it will also cease to be empty.

For the second case, we see the effect of the loss-feedback. When Q_k becomes full, the overall inflow rate into Q_1 (i.e. $\alpha_1(\theta; t)$) decreases due to the new additional loss rate from Q_k being fedback to the source. As a result, the inflow rate into Q_m will decrease and this

can be to the extent that there is a negative sign change in $A_m(\theta;t)$. Therefore, if Q_m was full, it will cease to be full.

Suppose instead Q_k ceases to be full so that the feedback decreases and as a result $\alpha_1(\theta;t)$ increases. This time, the inflow rate into Q_m will increase and can result in a positive sign change in $A_m(\theta;t)$ so that if Q_m was empty, it can cease to be empty.

5.5.2 The importance of event times

Now $\alpha_m(\theta;t)$ can switch between $\alpha_{m-1}(\theta;t)$ and $\beta_{m-1}(t)$ for $m = 2,\ldots,M$. Also $\alpha_1(\theta;t)$ can switch between $\sigma(t)$ and $\sigma(t) - \sum_{i\in FQ}\gamma_i(\theta;t)$ (where FQ is the set of indices of all queues that are full at time t). We see that at any given time the actual value of $\alpha_m(\theta;t), m = 1,\ldots,M$ is a function of the external processes, $\{\sigma(t)\}$ and $\{\beta_m(t)\}, m = 1,\ldots,M\}$, and possibly a function of the loss-feedback constant, c. If θ is $b_m, m = 1,\ldots,M$, then $\alpha_m(\theta;t), m = 1,\ldots,M$ will be independent of θ. However, the time at which $\alpha_m(\theta;t)$ switches depends on the buffer-occupancy process at node m, namely $\{x_m(\theta;t)\}$, $m = 1,\ldots,M$ and this in turn is dependent on θ. If the control parameter θ is the loss-feedback constant or some parameter of the service processes $\{\beta_m(t)\}, m = 1,\ldots,M\}$, the actual values of $\alpha_m(\theta;t), m = 1,\ldots,M$ will be dependent on θ.

Hence, in computing the IPA derivative estimators, the θ-dependence of the event times must be be considered.

5.5.3 θ-dependence of event times

For the analysis to follow, we shall encounter terms of the form $A_m(\theta;z_i)\frac{dz_i}{d\theta}$ where z_i is the time of event i. Here we try to establish which event types are θ-dependent and which are not. If the event is an exogenous event, then the time at which it occurs is the time for a jump in any of the external processes, and this is independent of θ. Therefore $\frac{dz_i}{d\theta} = 0 \Rightarrow A_m(\theta;z_i)\frac{dz_i}{d\theta} = 0$.

A type-1 endogenous event is dependent on θ since the control parameter θ dictates to a large extent the dynamics of the queue, Q_m, which will include the times at which the Q_m becomes full and when it becomes empty. Therefore $A_m(\theta;z_i)\frac{dz_i}{d\theta}$ can be a non-zero value. For the type-2 endogenous event that is due to a discontinuity in $\{\sigma(t)\}$ or $\{\beta_j(t)\}$,

$\frac{dz_i}{d\theta} = 0$. This is because the queue Q_m ceasing to be full or ceasing to be empty is due to an exogenous event in the external process which in turn is independent of θ. As a result $A_m(\theta; z_i)\frac{dz_i}{d\theta} = 0$. However, for the type-2 endogenous event that occurs when $A_m(\theta; t)$ is continuous at $t = z_i$, and $A_m(\theta; z_i) = 0$, so that the queue, Q_m, ceases to be full or ceases to be empty, $\frac{dz_i}{d\theta} \neq 0$ but the product $A_m(\theta; z_i)\frac{dz_i}{d\theta} = 0$ because $A_m(\theta; z_i) = 0$.

The determination of the θ-dependence of z_i for an induced event requires more careful deliberation. If the end of the boundary period at Q_m (i.e., it ceases to be full or it ceases to be empty), is caused by a type-1 endogenous event at another queue, then $\frac{dz_i}{d\theta} \neq 0$ so that $A_m(\theta; z_i)\frac{dz_i}{d\theta}$ is non-zero. However, if the end of the boundary period at Q_m is caused by a type-2 endogenous event occurring at another queue, say Q_k, then $\frac{dz_i}{d\theta}$ may or may not be equal to zero. In the case when that type-2 endogenous event is itself due to an exogenous event then $\frac{dz_i}{d\theta} = 0$. In the case when the type-2 endogenous event is itself due to its net inflow rate being continuous at $t = z_i$ so that $A_k(\theta; z_i) = 0$, then $\frac{dz_i}{d\theta} \neq 0$ and $A_m(\theta; z_i)\frac{dz_i}{d\theta} \neq 0$. But what shall be seen later is that the term we really must deal with for induced events is $(A_m(\theta; z_i^-) - A_m(\theta; z_i^+))\frac{dz_i}{d\theta}$ where the coefficient $(A_m(\theta; z_i^-) - A_m(\theta; z_i^+)) = \mu A_k(\theta; z_i)$ where μ is a constant. So for the latter case when $A_k(\theta; z_i) = 0$, the term $(A_m(\theta; z_i^-) - A_m(\theta; z_i^+))\frac{dz_i}{d\theta}$ disappears.

Based on this discussion on θ-dependence of event times, we see that for the first case in Section 5.5.1, that the time z_i at which the upstream Q_j becomes empty causing Q_m to cease to be full is dependent on θ, and for the second case in the same section, the time z_i at which the downstream queue, Q_k, becomes full causing Q_m to cease to be full is also dependent on θ.

We need to more carefully consider the case when the upstream queue, Q_j, upon ceasing to be empty, induces the queue in question, Q_m, to cease to be empty. We also need to carefully analyze the case when the downstream queue, Q_k, upon ceasing to be full, induces Q_m to cease-to-be empty. This is to ascertain the θ-dependence of those event times.

When Q_m ceases to be empty, it could be due to a local exogenous event or it can be due to Q_j, $(j = 1, \ldots, M,\ j < m)$ ceasing to be empty provided that the queues in between Q_j and Q_m are empty. However Q_j ceasing to be empty can also be due to a local exogenous

event or again to some upstream queue, Q_i, $i < j$, ceasing to be empty, an so on. Consider the extreme case when $j = 1$, so that it is Q_1 ceasing to be empty that induces Q_m to cease to be empty. This event at Q_1 could be due to an exogenous event local to Q_1 or it could be due to a downstream queue, Q_k, $k > m$ ceasing to be full, which in turn can be due to a local exogenous event at Q_k or can be due to some queue Q_i, $m < i < k$ becoming empty and this is an endogenous event. In those scenarios where the root inducing event is exogenous, the induced event time at Q_m will be independent of θ, i.e., $\frac{dz_i}{d\theta} = 0$. In those scenarios where the root inducing event is an endogenous event, $\frac{dz_i}{d\theta}$ is non-zero. Figure 18 depicts the case when the root event is an endogenous event which causes Q_m to cease to be empty.

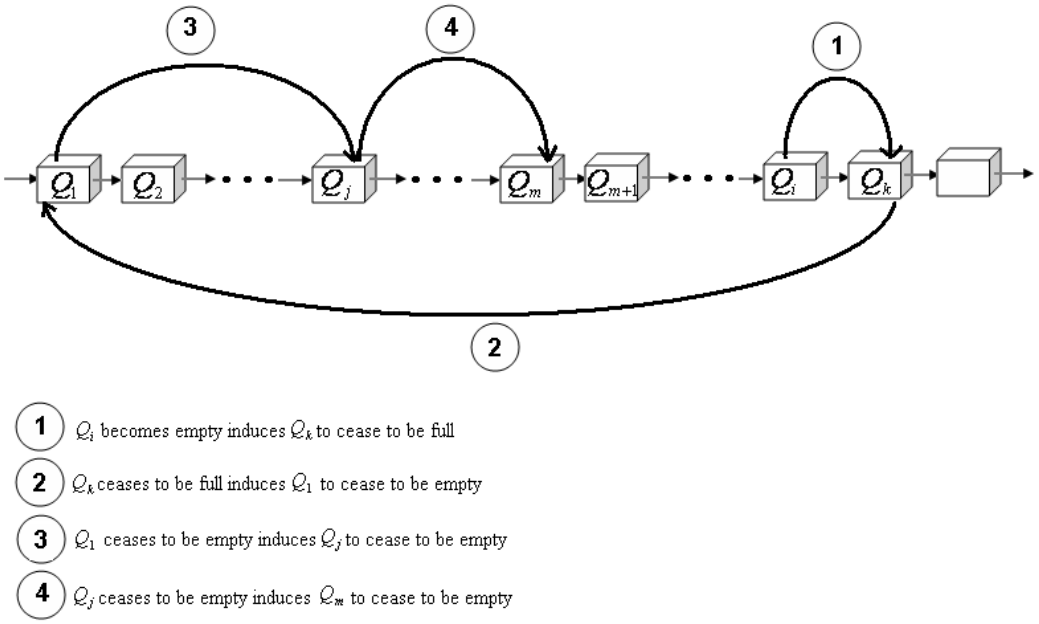


Figure 18: An example of an endogenous root inducing event

5.6 Assumptions

The sample derivatives $\frac{dL_m(\theta;T)}{d\theta}$ and $\frac{dQ_m(\theta;T)}{d\theta}$ depend to a large extent on the derivatives of the event-times with respect to θ. As a consequence, to identify the conditions under which the sample derivatives exist, we must determine the conditions under which the derivatives of the event times exist.

Assumptions:

(i) Realizations of the processes $\{\sigma(t)\}$, $\{\beta_m(t)\}, m = 1, \ldots, M\}$ are piecewise continuously differentiable on the interval $[0, T]$.

(ii) No boundary period consists of a single point.

(iii) For every $\theta \in \Theta$ w.p.1, no exogenous event co-occurs with any other exogenous event or with any type-1 endogenous event, and no type-1 endogenous event at one of the queues co-occurs with a type-1 endogenous event at the other queue.

(iv) It does not happen that $A_m(\theta; t) = 0$ for every t in an open sub-interval of any boundary period at Q_m. W.p.1 no two processes $\{\sigma(t)\}$ and $\{\beta_m(t)\}, m = 1, \ldots, M\}$ have identical values during an open sub-interval of $[0, T]$.

These assumptions are mild technical conditions. In the case of (iv) if $A_m(\theta; t) = \alpha_m(\theta; t) - \beta_m(t) = 0$ at some t, so that the derivative will not exist, the one-sided derivatives will exist and can be obtained through a finite difference analysis. Note that these are standard assumptions made throughout the IPA-SFM literature.

5.7 Switchover points

A switchover point[1] in $\alpha_m(\theta; t)$ for $m \geq 1$ (as seen in Eq. (101)) occurs when the following situations occur:

1. $Q_j, j < m$ enters into an EP while queues in between itself and Q_m, namely $Q_i, j+1 \leq i \leq m-1$ are empty, so that $\alpha_m(\theta; t)$ switches from $\beta_j(t)$ to $\alpha_j(\theta; t)$.

2. $Q_j, j < m$ leaves an EP while queues in between itself and Q_m, namely $Q_i, j+1 \leq i \leq m-1$ are empty, so that $\alpha_m(\theta; t)$ switches from $\alpha_j(\theta; t)$ to $\beta_j(t)$.

3. $Q_j, j \geq m$ enters into an FP while all upstream queues $Q_i, 1 \leq i \leq m-1$ are empty, so that $\alpha_m(\theta; t)$ switches from $\alpha_1(\theta; t^-)$ to $\alpha_1(\theta; t^+)$, where $\alpha_1(\theta; t^+) = \alpha_1(\theta; t^-) - c(\alpha_j(\theta; t^-) - \beta_j(t^-))$

[1]The terms "switchover point" and "active switchover point" were first defined in [78]

4. $Q_j, j \geq m$ leaves an FP while all upstream queues $Q_i, 1 \leq i \leq m-1$ are empty, so that $\alpha_m(\theta;t)$ switches from $\alpha_1(\theta;t^-)$ to $\alpha_1(\theta;t^+)$, where $\alpha_1(\theta;t^+) = \alpha_1(\theta;t^-) + c(\alpha_j(\theta;t^-) - \beta_j(t^-))$

A switchover point in $\alpha_m(\theta;t)$ is deemed *active* if it causes a jump in $\alpha_m(\theta;t)$ and if the derivative of its event time with respect to θ is non-zero. In other words, the time at which the switchover occurs is dependent of θ.

The first two types of switchover points (i.e (1) and (2)) are due to the tandem action of events occurring at queues upstream from Q_m. From the discussion in Section 5.5.3. for switchover points of (2) to be active the event at Q_j must have been induced, where the root inducing event was endogenous and would have been at some Q_k, $k > m$ downstream from Q_m. Let $s_{1,m,i}(\theta)$ denote the time of the i-th switchover point at Q_m that is of type (1). Let $s_{2,m,i}(\theta)$ denote the time of the i-th switchover point at Q_m that is of type (2). The next two types of switchover points (i.e. (3) and (4)) are due to the feedback action of loss events occurring at queues downstream from Q_m. Again, for switchover points of (4) to be active the event at Q_j must have been induced, where the root inducing event is itself endogenous and would have been at some Q_k, $m < k, j$, i.e., downstream from Q_m but upstream from Q_j. Let $r_{3,m,j}(\theta)$ denote the time of the j-th switchover point at Q_m that is of type (3), and let $r_{4,m,j}(\theta)$ denote the time of the j-th switchover point at Q_m that is of type (4).

Additionally, an active switchover point, $s_{1,m,i}(\theta)$, $s_{2,m,i}(\theta)$, $r_{3,m,j}(\theta)$ or $r_{4,m,j}(\theta)$, could have occurred in a BP or NBP of Q_m. To distinguish the switchover points by the type of interval they occurred, we define different index sets as follows:

$$\Psi_{s,m,n} := \{i : s_{1,m,i} \in B_{m,n}\} \tag{109}$$

$$\Upsilon_{s,m,n} := \{i : s_{2,m,i} \in B_{m,n}\} \tag{110}$$

$$\Psi^{\circ}_{s,m,n} := \{i : s_{1,m,i} \in (\tau_{m,n}, \varsigma_{m,n})\} \tag{111}$$

$$\Upsilon^{\circ}_{s,m,n} := \{i : s_{2,m,i} \in (\tau_{m,n}, \varsigma_{m,n})\} \tag{112}$$

$$\bar{\Psi}_{s,m,n} := \{i : s_{1,m,i} \in NB_{m,n}\} \tag{113}$$

$$\bar{\Upsilon}_{s,m,n} := \{i : s_{2,m,i} \in NB_{m,n}\} \tag{114}$$

$$\Psi_{r,m,n} := \{j : r_{3,m,j} \in B_{m,n}\} \tag{115}$$

$$\Upsilon_{r,m,n} := \{j : r_{4,m,j} \in B_{m,n}\} \tag{116}$$

$$\Psi^{\circ}_{r,m,n} := \{j : r_{3,m,j} \in (\tau_{m,n}, \varsigma_{m,n})\} \tag{117}$$

$$\Upsilon^{\circ}_{r,m,n} := \{j : r_{4,m,j} \in (\tau_{m,n}, \varsigma_{m,n})\} \tag{118}$$

$$\bar{\Psi}_{r,m,n} := \{j : r_{3,m,j} \in NB_{m,n}\} \tag{119}$$

$$\bar{\Upsilon}_{r,m,n} := \{j : r_{4,m,j} \in NB_{m,n}\} \tag{120}$$

There are situations when an active switchover point induces the close of an BP at Q_m, hence the need to define the sets $\Psi_{(\cdot),m,n}$ and $\Upsilon_{(\cdot),m,n}$ on the closed interval $B_{m,n} = [\tau_{m,n}, \varsigma_{m,n}]$ and the sets $\Psi^{\circ}_{(\cdot),m,n}$ and $\Upsilon^{\circ}_{(\cdot),m,n}$ on the open interval $(\tau_{m,n}, \varsigma_{m,n})$. Similar definitions were made in [78].

5.8 The recursive relation among queues

Define the term $\xi_m(\theta; z), m = 1, \ldots, M$ for every point z which is the starting point of a boundary period at Q_m as:

$$\xi_m(\theta; z) = A_m(\theta; z^-)\frac{dz}{d\theta} \tag{121}$$

$\xi_m(\theta; z)$ is a recursive equation in the time domain that characterizes the interplay among the queues in the tandem so that $\xi_m(\theta; z_a)$ depends on $\xi_j(\theta; z_b), j = 1, \ldots, M \quad m \neq j$ where $z_b < z_a$ and z_b, z_a can be the starting times of the respective boundary periods at the different queues $Q_j, j = 1, \ldots, M$. This recursion can be observed from the sample path through fairly simple counting processes.

We will express the IPA sample derivatives $\frac{dL_m(\theta;T)}{d\theta}$ and $\frac{dQ_m(\theta;T)}{d\theta}$ in terms of $\xi_j(\theta; s_{1,m,i})$, and $\xi_j(\theta; r_{3,m,k})$ $j = 1, \ldots, M,\ i = 1, \ldots,\ k = 1, \ldots,\ m \neq j$.

We wish to also define the following term:

$$\rho_{m,n} = \begin{cases} \pi_{m,n} & \text{if } \theta \equiv b_m \\ 0 & \text{otherwise} \end{cases} \tag{122}$$

where

$$\pi_{m,n} = \begin{cases} 1 & \text{if } NB_{m,n} = EF \\ -1 & \text{if } NB_{m,n} = FE \\ 0 & \text{if } NB_{m,n} = EE \text{ or if } NB_{m,n} = FF \end{cases} \tag{123}$$

We will show that for every $n = 1, \ldots, N_m$, $m = 1, \ldots, M$, if

(i) The buffer capacity of Q_m, i.e. b_m, is the control parameter, θ, then

$$\xi_m(\theta; \tau_{m,n}) = \rho_{m,n} + A_m(\theta; \varsigma^+_{m,n-1})\frac{d\varsigma_{m,n-1}}{d\theta} - \sum_{j=m+1}^{M} \sum_{\substack{i \in \bar{\Psi}_{r,m,n} \\ j \to i}} c\xi_j(\theta; r_{3,m,i})$$

$$+ \sum_{j=1}^{m-1} \sum_{\substack{i \in \bar{\Psi}_{s,m,n} \\ j \to i}} \xi_j(\theta; s_{1,m,i})$$

$$
\begin{aligned}
&- \sum_{l=m+1}^{M} \sum_{\substack{j=1 \\ l\to j}}^{m-1} \sum_{\substack{i\in\bar{\Upsilon}_{s,m,n} \\ j\to i}} \frac{A_j(\theta; s^-_{2,m,i})}{A_l(\theta; s_{2,m,i})}\xi_l(\theta; s_{2,m,i}) \\
&+ \sum_{l=m+1}^{M} \sum_{\substack{j=m+1 \\ l\to j}}^{M} \sum_{\substack{i\in\bar{\Upsilon}_{r,m,n} \\ j\to i}} c\frac{A_j(\theta; r^-_{4,m,i})}{A_l(\theta; r_{4,m,i})}\xi_l(\theta; r_{4,m,i}) \qquad (124)
\end{aligned}
$$

where the notation "$l \to j$" means that Q_l is the root inducer of the event at Q_j, and the notation "$j \to i$" means that Q_j causes the active switchover point $s_{1,m,i}$ (or $r_{3,m,i}$) at Q_m.

(ii) The loss-feedback constant, c, is the control parameter, θ, then

$$
\begin{aligned}
\xi_m(\theta;\tau_{m,n}) \;=\;& \rho_{m,n} - L_{net}(\theta, \varsigma_{m,n-1}, \tau_{m,n}) + A_m(\theta;\varsigma^+_{m,n-1})\frac{d\varsigma_{m,n-1}}{d\theta} \\
&- \sum_{j=m+1}^{M} \sum_{\substack{i\in\bar{\Psi}_{r,m,n} \\ j\to i}} c\xi_j(\theta; r_{3,m,i}) + \sum_{j=1}^{m-1} \sum_{\substack{i\in\bar{\Psi}_{s,m,n} \\ j\to i}} \xi_j(\theta; s_{1,m,i}) \\
&- \sum_{l=m+1}^{M} \sum_{\substack{j=1 \\ l\to j}}^{m-1} \sum_{\substack{i\in\bar{\Upsilon}_{s,m,n} \\ j\to i}} \frac{A_j(\theta; s^-_{2,m,i})}{A_l(\theta; s_{2,m,i})}\xi_l(\theta; s_{2,m,i}) \\
&+ \sum_{l=m+1}^{M} \sum_{\substack{j=m+1 \\ l\to j}}^{M} \sum_{\substack{i\in\bar{\Upsilon}_{r,m,n} \\ j\to i}} c\frac{A_j(\theta; r^-_{4,m,i})}{A_l(\theta; r_{4,m,i})}\xi_l(\theta; r_{4,m,i}) \qquad (125)
\end{aligned}
$$

Where in both cases:

$$
A_m(\theta;\varsigma^+_{m,n-1})\frac{d\varsigma_{m,n-1}}{d\theta} = \begin{cases} \frac{A_m(\theta;\varsigma^+_{m,n-1})}{A_j(\theta;\varsigma^-_{m,n-1})}\xi_j(\theta;\varsigma_{m,n-1}) & \text{if } n-1 \in \Phi_m \text{ and if } Q_j \text{ at } \varsigma_{m,n-1} \\ & \text{caused the end of } B_{m,n-1} \\ 0 & \text{otherwise} \end{cases} \qquad (126)
$$

Also, $L_{net}(\theta, \varsigma_{m,n-1}, \tau_{m,n})$ is the total loss volume of all the queues in the tandem when $Q_i, i = 1, \ldots, m-1$ are empty during the period $(\varsigma_{m,n-1}, \tau_{m,n})$.

Consider the NBP $NB_{m,n}$ at Q_m:

$$x_m(\theta;t)|_{\varsigma_{m,n-1}}^{\tau_{m,n}} = x_m(\theta;\tau_{m,n}) - x_m(\theta;\varsigma_{m,n-1}) \quad = \quad \begin{cases} b_m & \text{if } NBP \text{ is } EF \\ -b_m & \text{if } NBP \text{ is } FE \\ 0 & \text{if } NBP \text{ is } EE \text{ or } FF \end{cases} \tag{127}$$

Also

$$\begin{aligned} x_m(\theta;t)|_{\varsigma_{m,n-1}}^{\tau_{m,n}} &= \int_{\varsigma_{m,n-1}}^{\tau_{m,n}} \frac{dx_m(\theta,t)}{dt}\,dt & (128)\\ &= \int_{\varsigma_{m,n-1}}^{\tau_{m,n}} A_m(\theta;t)\,dt & (129)\\ &= \int_{\varsigma_{m,n-1}}^{z_{m,n,1}} A_m(\theta;t)\,dt + \sum_{i=1}^{K_{m,n}-1} \int_{z_{m,n,i}}^{z_{m,n,i+1}} A_m(\theta;t)\,dt \\ &\quad + \int_{z_{m,n,K_{m,n}}}^{\tau_{m,n}} A_m(\theta;t)\,dt & (130) \end{aligned}$$

where $z_{m,n,i}, i = 1,\ldots,K_{m,n}$ are the time-points in the interior of $NB_{m,n}$ at which $A_m(\theta;t)$ is discontinuous.

Taking derivatives of Eq. (127) with respect to θ:

$$\frac{dx_m(\theta;t)}{d\theta} \quad = \quad \rho_{m,n} \tag{131}$$

And taking derivatives of Eq. (130) with respect to θ and using the Leibniz rule:

$$\begin{aligned} \tfrac{dx_m(\theta;t)}{d\theta} &= \tfrac{d}{d\theta}\left[\int_{\varsigma_{m,n-1}}^{z_{m,n,1}} A_m(\theta;t)\,dt + \sum_{i=1}^{K_{m,n}-1}\int_{z_{m,n,i}}^{z_{m,n,i+1}} A_m(\theta;t)\,dt + \int_{z_{m,n,K_{m,n}}}^{\tau_{m,n}} A_m(\theta;t)\,dt\right] \\ &= \int_{\varsigma_{m,n-1}}^{z_{m,n,1}} \tfrac{dA_m(\theta;t)}{d\theta}\,dt + A_m(\theta;z_{m,n,1}^-)\tfrac{dz_{m,n,1}}{d\theta} - A_m(\theta;\varsigma_{m,n-1}^+)\tfrac{d\varsigma_{m,n-1}}{d\theta} \\ &\quad + \sum_{i=1}^{K_{m,n}-1}\left[\int_{z_{m,n,i}}^{z_{m,n,i+1}} \tfrac{dA_m(\theta;t)}{d\theta}\,dt + A_m(\theta;z_{m,n,i+1}^-)\tfrac{dz_{m,n,i+1}}{d\theta} - A_m(\theta;z_{m,n,i}^+)\tfrac{dz_{m,n,i}}{d\theta}\right] \\ &\quad + \int_{z_{m,n,K_{m,n}}}^{\tau_{m,n}} \tfrac{dA_m(\theta;t)}{d\theta}\,dt + A_m(\theta;\tau_{m,n}^-)\tfrac{d\tau_{m,n}}{d\theta} - A_m(\theta;z_{m,n,K_{m,n}}^+)\tfrac{dz_{m,n,K_{m,n}}}{d\theta} \end{aligned} \tag{132}$$

5.8.1 Buffer capacity, b_m, as the control parameter, θ

Since $A_m(\theta;t) = \alpha_m(\theta;t) - \beta_m(t)$ and $\alpha_m(\theta;t)$ is a function of $\sigma(t)$, $\beta_j(t), j = 1\ldots,M$ and c which in turn are all independent of θ, then the value of $A_m(\theta;t)$ in between queue

event-times is independent of θ . Therefore, $\frac{dA_m(\theta;t)}{d\theta} = 0 \Rightarrow \int_a^b \frac{dA_m(\theta;t)}{d\theta}\, dt = 0$ where a and b are two consecutive queue event-times. Eq. (132) then becomes:

$$\begin{aligned} \frac{dx_m(\theta;t)}{d\theta} &= A_m(\theta;\tau^-_{m,n})\frac{d\tau_{m,n}}{d\theta} + \sum_{i=1}^{K_{m,n}} \left(A_m(\theta;z^-_{m,n,i}) - A_m(\theta;z^+_{m,n,i})\right)\frac{dz_{m,n,i}}{d\theta} \\ &- A_m(\theta;\varsigma^+_{m,n-1})\frac{d\varsigma_{m,n-1}}{d\theta} \end{aligned} \tag{133}$$

Equating Eq. (131) to Eq. (133):

$$\begin{aligned} A_m(\theta;\tau^-_{m,n})\frac{d\tau_{m,n}}{d\theta} &= \rho_{m,n} + A_m(\theta;\varsigma^+_{m,n-1})\frac{d\varsigma_{m,n-1}}{d\theta} \\ &\quad -\sum_{i=1}^{K_{m,n}} \left(A_m(\theta;z^-_{m,n,i}) - A_m(\theta;z^+_{m,n,i})\right)\frac{dz_{m,n,i}}{d\theta} \end{aligned} \tag{134}$$

In Eq. (134), the term in $\frac{dz_{m,n,i}}{d\theta}$ will generally be non-zero if it corresponds to an active switchover point at Q_m. Consider those active switchover points that correspond to the tandem action of queues upstream from Q_m, i.e. $s_{1,m,i}(\theta)$ and $s_{2,m,i}(\theta)$, then for $j < m, j = 1\ldots, M$ and for $s_{1,m,i}(\theta)$:

$$A_m(\theta;s^-_{1,m,i}) = \beta_j(s_{1,m,i}) - \beta_m(s_{1,m,i}) \tag{135}$$

$$A_m(\theta;s^+_{1,m,i}) = \alpha_j(\theta;s_{1,m,i}) - \beta_m(s_{1,m,i}) \tag{136}$$

$$\Rightarrow A_m(\theta;s^-_{1,m,i}) - A_m(\theta;s^+_{1,m,i}) = -(\alpha_j(\theta;s_{1,m,i}) - \beta_j(s_{1,m,i})) = -A_j(\theta;s_{1,m,i}) \tag{137}$$

Note that at $t = s_{1,m,i}$, $A_j(\theta;t)$ is continuous so that $A_j(\theta;s^-_{1,m,i}) = A_j(\theta;s^+_{1,m,i}) = A_j(\theta;s_{1,m,i})$. For $s_{2,m,i}(\theta)$:

$$A_m(\theta;s^-_{1,m,i}) = \alpha_j(\theta;s^-_{2,m,i}) - \beta_m(s_{2,m,i}) \tag{138}$$

$$A_m(\theta;s^+_{1,m,i}) = \beta_j(s_{2,m,i}) - \beta_m(s_{2,m,i}) \tag{139}$$

$$\Rightarrow A_m(\theta;s^-_{1,m,i}) - A_m(\theta;s^+_{1,m,i}) = (\alpha_j(\theta;s^-_{2,m,i}) - \beta_j(s_{2,m,i})) = A_j(\theta;s^-_{2,m,i}) \tag{140}$$

For those switchover points that correspond to the feedback action of queues that are downstream from Q_m, i.e. $r_{3,m,i}(\theta)$ and $r_{4,m,i}(\theta)$, we have for $j > m, j = 1\ldots, M$ and for $r_{3,m,i}(\theta)$:

$$A_m(\theta;r^-_{3,m,i}) = \alpha_1(\theta;r^-_{3,m,i}) - \beta_m(r_{3,m,i}) \tag{141}$$

$$
\begin{aligned}
A_m(\theta; r^+_{3,m,i}) &= \alpha_1(\theta; r^+_{3,m,i}) - \beta_m(r_{3,m,i}) && (142)\\
\Rightarrow A_m(\theta; r^-_{3,m,i}) - A_m(\theta; r^+_{3,m,i}) &= \alpha_1(\theta; r^-_{3,m,i}) - \alpha_1(\theta; r^+_{3,m,i}) && (143)\\
&= c\gamma_j(\theta; r_{3,m,i}) && (144)\\
&= c(\alpha_j(\theta; r_{3,m,i}) - \beta_j(r_{3,m,i})) && (145)\\
&= cA_j(\theta; r_{3,m,i}) && (146)
\end{aligned}
$$

Again note that at $t = r_{3,m,i}$, $A_j(\theta; t)$ is continuous since there are no changes in $\alpha_j(\theta; t)$ and $\beta_j(t)$ at that time. Hence $A_j(\theta; r^-_{3,m,i}) = A_j(\theta; r^+_{3,m,i}) = A_j(\theta; r_{3,m,i})$. By similar analysis we see that for $r_{4,m,i}(\theta)$:

$$
\begin{aligned}
A_m(\theta; r^-_{4,m,i}) &= \alpha_1(\theta; r^-_{4,m,i}) - \beta_m(r_{4,m,i}) && (147)\\
A_m(\theta; r^+_{4,m,i}) &= \alpha_1(\theta; r^+_{4,m,i}) - \beta_m(r_{4,m,i}) && (148)\\
\Rightarrow A_m(\theta; r^-_{4,m,i}) - A_m(\theta; r^+_{4,m,i}) &= \alpha_1(\theta; r^-_{4,m,i}) - \alpha_1(\theta; r^+_{4,m,i}) && (149)\\
&= -c\gamma_j(\theta; r_{4,m,i}) && (150)\\
&= -c(\alpha_j(\theta; r_{4,m,i}) - \beta_j(r_{4,m,i})) && (151)\\
&= -cA_j(\theta; r_{4,m,i}) && (152)
\end{aligned}
$$

Substituting Eq. (137), Eq. (140), Eq. (146) and Eq. (152) into Eq. (134) leads to:

$$
\begin{aligned}
A_m(\theta; \tau_{m,n})\frac{d\tau_{m,n}}{d\theta} &= \rho_{m,n} + A_m(\theta; \varsigma^+_{m,n-1})\frac{d\varsigma_{m,n-1}}{d\theta} - \sum_{j=m+1}^{M} \sum_{\substack{i \in \bar{\Psi}_{r,m,n} \\ j \to i}} cA_j(\theta; r_{3,m,i})\frac{dr_{3,m,i}}{d\theta}\\
&\quad + \sum_{j=1}^{m-1} \sum_{\substack{i \in \bar{\Psi}_{s,m,n} \\ j \to i}} A_j(\theta; s_{1,m,i})\frac{ds_{1,m,i}}{d\theta}\\
&\quad - \sum_{l=m+1}^{M} \sum_{\substack{j=1 \\ l \to j}}^{m-1} \sum_{\substack{i \in \bar{\Upsilon}_{s,m,n} \\ j \to i}} \frac{A_j(\theta; s^-_{2,m,i})}{A_l(\theta; s_{2,m,i})} A_l(\theta; s_{2,m,i})\frac{ds_{2,m,i}}{d\theta}
\end{aligned}
$$

$$+ \sum_{l=m+1}^{M} \sum_{\substack{j=m+1 \\ l \to j}}^{M} \sum_{\substack{i \in \bar{\Upsilon}_{r,m,n} \\ j \to i}} c \frac{A_j(\theta; r^-_{4,m,i})}{A_l(\theta; r_{4,m,i})} A_l(\theta; r_{4,m,i}) \frac{dr_{4,m,i}}{d\theta} \qquad (153)$$

Using the definition in Eq. (121) and the continuity in $A_j(\theta; s_{1,m,i})$ and $A_j(\theta; r_{3,m,i})$, Eq. (153) becomes

$$\begin{aligned}
\xi_m(\theta; \tau_{m,n}) &= \rho_{m,n} + A_m(\theta; \varsigma^+_{m,n-1}) \frac{d\varsigma_{m,n-1}}{d\theta} - \sum_{j=m+1}^{M} \sum_{\substack{i \in \bar{\Psi}_{r,m,n} \\ j \to i}} c\xi_j(\theta; r_{3,m,i}) \\
&+ \sum_{j=1}^{m-1} \sum_{\substack{i \in \bar{\Psi}_{s,m,n} \\ j \to i}} \xi_j(\theta; s_{1,m,i}) \\
&- \sum_{l=m+1}^{M} \sum_{\substack{j=1 \\ l \to j}}^{m-1} \sum_{\substack{i \in \bar{\Upsilon}_{s,m,n} \\ j \to i}} \frac{A_j(\theta; s^-_{2,m,i})}{A_l(\theta; s_{2,m,i})} \xi_l(\theta; s_{2,m,i}) \\
&+ \sum_{l=m+1}^{M} \sum_{\substack{j=m+1 \\ l \to j}}^{M} \sum_{\substack{i \in \bar{\Upsilon}_{r,m,n} \\ j \to i}} c \frac{A_j(\theta; r^-_{4,m,i})}{A_l(\theta; r_{4,m,i})} \xi_l(\theta; r_{4,m,i}) \qquad (154)
\end{aligned}$$

Now, $A_m(\theta; \varsigma^+_{m,n-1}) \frac{d\varsigma_{m,n-1}}{d\theta} = 0$ except if it was induced by an active switchover point $s_{1,m,i}(\theta)$ or $r_{3,m,i}(\theta)$.

$$\begin{aligned}
A_m(\theta; \varsigma^+_{m,n-1}) \frac{d\varsigma_{m,n-1}}{d\theta} &= \frac{A_m(\theta; \varsigma^+_{m,n-1})}{A_j(\theta; \varsigma^-_{m,n-1})} \left(A_j(\theta; \varsigma^-_{m,n-1}) \frac{d\varsigma_{m,n-1}}{d\theta} \right) \qquad (155) \\
&= \frac{A_m(\theta; \varsigma^+_{m,n-1})}{A_j(\theta; \varsigma^-_{m,n-1})} \xi_j(\theta; \varsigma_{m,n-1}) \qquad (156)
\end{aligned}$$

5.8.2 Loss-feedback constant, c, as the control parameter, θ

Consider again the term $\int_a^b \frac{dA_m(\theta;t)}{d\theta}\,dt$ in Eq. (132). In this case, the value of $\frac{dA_m(\theta;t)}{d\theta}$ between event-times a and b can be non-zero. For $t \in (a,b)$,

$$A_m(\theta;t) = \begin{cases} \beta_j(t) - \beta_m(t) & \text{if } j < m \text{ and } x_j(\theta;t) > 0 \\ \alpha_1(\theta;t) - \beta_m(t) & \text{if all } Q_i, i = 1,\ldots,m-1 \text{ are empty} \end{cases} \tag{157}$$

$$\Rightarrow \frac{dA_m(\theta;t)}{d\theta} = \begin{cases} 0 & \text{if } j < m \text{ and } x_j(\theta;t) > 0 \\ \frac{d\alpha_1(\theta;t)}{d\theta} & \text{if all } Q_i, i = 1,\ldots,m-1 \text{ are empty} \end{cases} \tag{158}$$

Now

$$\alpha_1(\theta;t) = \begin{cases} \sigma(t) & \text{if there are no full queues} \\ \sigma(t) - c\sum\gamma_i(\theta;t) & \text{if any } Q_i, i = m+1,\ldots,M \text{ are full} \\ \frac{\sigma(t) - c\sum\gamma_i(\theta;t) + c\beta_m(t)}{1+c} & \text{if any } Q_i, i = m+1,\ldots,M \text{ as well as } Q_m \text{ are full} \end{cases} \tag{159}$$

$$\Rightarrow \frac{d\alpha_1(\theta;t)}{d\theta} = \begin{cases} 0 & \text{if there are no full queues} \\ -\sum\gamma_i(\theta;t) & \text{if any } Q_i, i = m+1,\ldots,M \text{ are full} \\ -\frac{\sigma(t) - \beta_m(t) + \sum\gamma_i(\theta;t)}{(1+c)^2} & \text{if any } Q_i, i = m+1,\ldots,M \text{ as well as } Q_m \text{ are full} \end{cases} \tag{160}$$

Our current analysis pertains to an NBP in Q_m, as a result there will be no losses in Q_m during this time. Eq. (160) shows that for an NBP, the only time $\frac{dA_m(\theta;t)}{d\theta}$ will be non-zero is when all upstream queues from Q_m, namely $Q_i, i = 1,\ldots,m-1$ are empty, and there are losses in one or some of the downstream queues $Q_j, j = m+1,\ldots,M$. This non-zero value equates negative sum of the loss-rates at these downstream queues. Essentially, $\int_a^b \frac{dA_m(\theta;t)}{d\theta}\,dt$ is the total loss-volume in the network during the times that all upstream queues $Q_i, i = 1,\ldots,m-1$ are empty within the period (a,b). Let $L_{net}(\theta,a,b)$ denote such a loss-volume, i.e.

$$L_{net}(\theta,a,b) \equiv \int_a^b \frac{dA_m(\theta;t)}{d\theta}\,dt \tag{161}$$

The remainder of the analysis follows the same as that when b_m was the control parameter θ, as from Eq. (133) to Eq. (156), except for the inclusion of Eq. (161) so that

$$\xi_m(\theta;\tau_{m,n}) = \rho_{m,n} - L_{net}(\theta,\varsigma_{m,n-1},\tau_{m,n}) + A_m(\theta;\varsigma^+_{m,n-1})\frac{d\varsigma_{m,n-1}}{d\theta} \tag{162}$$

$$
\begin{aligned}
&- \sum_{j=m+1}^{M} \sum_{\substack{i\in\bar{\Psi}_{r,m,n} \\ j\to i}} c\xi_j(\theta; r_{3,m,i}) + \sum_{j=1}^{m-1} \sum_{\substack{i\in\bar{\Psi}_{s,m,n} \\ j\to i}} \xi_j(\theta; s_{1,m,i}) \\
&- \sum_{l=m+1}^{M} \sum_{\substack{j=1 \\ l\to j}}^{m-1} \sum_{\substack{i\in\bar{\Upsilon}_{s,m,n} \\ j\to i}} \frac{A_j(\theta; s^-_{2,m,i})}{A_l(\theta; s_{2,m,i})}\xi_l(\theta; s_{2,m,i}) \\
&+ \sum_{l=m+1}^{M} \sum_{\substack{j=m+1 \\ l\to j}}^{M} \sum_{\substack{i\in\bar{\Upsilon}_{r,m,n} \\ j\to i}} c\frac{A_j(\theta; r^-_{4,m,i})}{A_l(\theta; r_{4,m,i})}\xi_l(\theta; r_{4,m,i})
\end{aligned} \tag{163}
$$

where

$$
A_m(\theta; \varsigma^+_{m,n-1})\frac{d\varsigma_{m,n-1}}{d\theta} = \begin{cases} \frac{A_m(\theta;\varsigma^+_{m,n-1})}{A_j(\theta;\varsigma^-_{m,n-1})}\xi_j(\theta; \varsigma_{m,n-1}) & \text{if } n-1 \in \Phi_m \text{ and if } Q_j \text{ at } \varsigma_{m,n-1} \\ & \text{caused the end of } B_{m,n-1} \\ 0 & \text{otherwise} \end{cases} \tag{164}
$$

5.9 IPA loss derivative, $\frac{dL(\theta,T)}{d\theta}$

Define $\mathcal{F}_m$ as the set of all indices of BPs that are FPs at node m, i.e.

$$\mathcal{F}_m := \{n : x_m(\theta;t) = b_m, \text{for all } t \in B_{m,n},\ n = 1,\ldots,N_m\} \tag{165}$$

Therefore, the total loss volume at node m, i.e. $L_m(\theta;T)$ is given by:

$$L_m(\theta;T) = \sum_{n\in\mathcal{F}_m} \int_{\tau_{m,n}}^{\varsigma_{m,n}} \gamma_m(\theta;t)\,dt = \sum_{n\in\mathcal{F}_m} \int_{\tau_{m,n}}^{\varsigma_{m,n}} A_m(\theta;t)\,dt \tag{166}$$

Let $z_{m,n,i}, i = 1,\ldots,J_{m,n}$ be the time-points in the interior of the BP at node m (i.e. $n \in \mathcal{F}_m$) at which $A_m(\theta;t)$ is discontinuous. Therefore Eq. (166) becomes

$$\begin{aligned} L_m(\theta;T) &= \sum_{n\in\mathcal{F}_m} \Bigg[\int_{\tau_{m,n}}^{z_{m,n,1}} A_m(\theta;t)\,dt + \sum_{i=1}^{J_{m,n}-1} \int_{z_{m,n,i}}^{z_{m,n,i+1}} A_m(\theta;t)\,dt \\ &\quad + \int_{z_{m,n,J_{m,n}}}^{\varsigma_{m,n}} A_m(\theta;t)\,dt \Bigg] \end{aligned} \tag{167}$$

Taking derivatives of Eq. (168) with respect to θ:

$$\begin{aligned} \tfrac{dL_m(\theta;T)}{d\theta} &= \textstyle\sum_{n\in\mathcal{F}_m} \Big[\frac{d}{d\theta} \Big\{ \int_{\tau_{m,n}}^{z_{m,n,1}} A_m(\theta;t)\,dt + \sum_{i=1}^{J_{m,n}-1} \int_{z_{m,n,i}}^{z_{m,n,i+1}} A_m(\theta;t)\,dt + \int_{z_{m,n,J_{m,n}}}^{\varsigma_{m,n}} A_m(\theta;t)\,dt \Big\} \Big] \\ &= \textstyle\sum_{n\in\mathcal{F}_m} \Big[\int_{\tau_{m,n}}^{z_{m,n,1}} \frac{dA_m(\theta;t)}{d\theta}\,dt + A_m(\theta;z_{m,n,1}^-)\frac{dz_{m,n,1}}{d\theta} - A_m(\theta;\tau_{m,n}^+)\frac{d\tau_{m,n}}{d\theta} \\ &\quad + \sum_{i=1}^{J_{m,n}-1} \Big\{ \int_{z_{m,n,i}}^{z_{m,n,i+1}} \frac{dA_m(\theta;t)}{d\theta}\,dt + A_m(\theta;z_{m,n,i+1}^-)\frac{dz_{m,n,i+1}}{d\theta} - A_m(\theta;z_{m,n,i}^+)\frac{dz_{m,n,i}}{d\theta} \Big\} \\ &\quad + \int_{z_{m,n,J_{m,n}}}^{\varsigma_{m,n}} \frac{dA_m(\theta;t)}{d\theta}\,dt + A_m(\theta;\varsigma_{m,n}^-)\frac{d\varsigma_{m,n}}{d\theta} - A_m(\theta;z_{m,n,J_{m,n}}^+)\frac{dz_{m,n,J_{m,n}}}{d\theta} \Big] \end{aligned} \tag{168}$$

5.9.1 Buffer capacity, b_m, as the control parameter, θ

Remember the value of $A_m(\theta;t)$ in between queue event-times is independent of θ . Therefore, $\frac{dA_m(\theta;t)}{d\theta} = 0 \Rightarrow \int_a^b \frac{dA_m(\theta;t)}{d\theta}\,dt = 0$ where a and b are two consecutive queue event-times. Eq. (168) then becomes:

$$\begin{aligned} \tfrac{dL_m(\theta;T)}{d\theta} &= \textstyle\sum_{n\in\mathcal{F}_m} \Big[A_m(\theta;\varsigma_{m,n}^-)\frac{d\varsigma_{m,n}}{d\theta} + \sum_{i=1}^{K_{m,n}} \Big(A_m(\theta;z_{m,n,i}^-) - A_m(\theta;z_{m,n,i}^+) \Big) \frac{dz_{m,n,i}}{d\theta} \\ &- A_m(\theta;\tau_{m,n}^+)\frac{d\tau_{m,n}}{d\theta} \Big] \end{aligned} \tag{169}$$

The term in $\frac{dz_{m,n,i}}{d\theta}$ in Eq. (169) corresponds to an active switchover point at Q_m. Therefore, consider those active switchover points that correspond to the tandem action of queues upstream from Q_m, i.e., $s_{1,m,i}(\theta)$ and $s_{2,m,i}(\theta)$, $j < m, j = 1 \ldots, M$. For $s_{1,m,i}(\theta)$:

$$\begin{aligned} A_m(\theta; s^-_{1,m,i}) &= \beta_j(s_{1,m,i}) - \beta_m(s_{1,m,i}) && (170)\\ A_m(\theta; s^+_{1,m,i}) &= \alpha_j(\theta; s^+_{1,m,i}) - \beta_m(s_{1,m,i}) && (171)\\ \Rightarrow A_m(\theta; s^-_{1,m,i}) - A_m(\theta; s^+_{1,m,i}) &= -(\alpha_j(\theta; s^+_{1,m,i}) - \beta_j(s_{1,m,i})) && (172)\\ &= -A_j(\theta; s^+_{1,m,i}) && (173) \end{aligned}$$

Now if any queues upstream from Q_j is non-empty, i.e., Q_i, $1 \leq i < j$, then

$$A_j(\theta; s^+_{1,m,i}) = A_j(\theta; s^-_{1,m,i}) \quad (174)$$

However, if all queues upstream from Q_j are empty, then

$$A_j(\theta; s^+_{1,m,i}) = \frac{1}{1+c} A_j(\theta; s^-_{1,m,i}) \quad (175)$$

This is so for the latter case because:

$$\begin{aligned} A_j(\theta; s^-_{1,m,i}) &= \alpha_j(\theta; s^-_{1,m,i}) - \beta_j(s_{1,m,i}) && (176)\\ &= \sigma(s_{1,m,i}) - c(\beta_j(s_{1,m,i}) - \beta_m(s_{1,m,i})) - c\sum_{k=m+1}^{M} \gamma_k(\theta; s_{1,m,i}) - \beta_j(s_{1,m,i}) && (177)\\ &= \sigma(s_{1,m,i}) - (1+c)\beta_j(s_{1,m,i}) + c\beta_m(s_{1,m,i}) - c\sum_{k=m+1}^{M} \gamma_k(\theta; s_{1,m,i}) && (178)\\ A_j(\theta; s^+_{1,m,i}) &= \alpha_j(\theta; s^+_{1,m,i}) - \beta_j(s_{1,m,i}) && (179)\\ &= \frac{\sigma(s_{1,m,i}) + c\beta_m(s_{1,m,i}) - c\sum_{k=m+1}^{M} \gamma_k(\theta; s_{1,m,i})}{1+c} - \beta_j(s_{1,m,i}) && (180)\\ &= \frac{\sigma(s_{1,m,i}) - (1+c)\beta_j(s_{1,m,i}) + c\beta_m(s_{1,m,i}) - c\sum_{k=m+1}^{M} \gamma_k(\theta; s_{1,m,i})}{1+c} && (181)\\ &= \frac{1}{1+c} A_j(\theta; s^-_{1,m,i}) && (182) \end{aligned}$$

Hence, we define the following term:

$$\kappa_j = \begin{cases} \frac{1}{1+c} & \text{if all } Q_i, i = 1, \ldots, j-1 \text{ are empty} \\ 1 & \text{otherwise} \end{cases} \quad (183)$$

so that

$$\Rightarrow A_m(\theta; s^-_{1,m,i}) - A_m(\theta; s^+_{1,m,i}) = -\kappa_j A_j(\theta; s^-_{1,m,i}) \quad (184)$$

For $s_{2,m,i}(\theta)$:

$$\begin{aligned} A_m(\theta; s^-_{2,m,i}) &= \alpha_j(\theta; s^-_{2,m,i}) - \beta_m(s_{2,m,i}) && (185)\\ A_m(\theta; s^+_{2,m,i}) &= \beta_j(s_{2,m,i}) - \beta_m(s_{2,m,i}) && (186)\\ \Rightarrow A_m(\theta; s^-_{1,m,i}) - A_m(\theta; s^+_{1,m,i}) &= \alpha_j(\theta; s^-_{1,m,i}) - \beta_j(s_{1,m,i}) && (187)\\ &= A_j(\theta; s^+_{1,m,i}) && (188) \end{aligned}$$

Consider those switchover points that correspond to the feedback action of queues that are downstream from Q_m, i.e., $r_{3,m,i}(\theta)$ and $r_{4,m,i}(\theta)$. For $r_{3,m,i}(\theta)$ with $j > m, j = 1 \ldots, M$

$$\begin{aligned} A_m(\theta; r^-_{3,m,i}) &= \alpha_1(\theta; r^-_{3,m,i}) - \beta_m(r_{3,m,i}) && (189)\\ A_m(\theta; r^+_{3,m,i}) &= \alpha_1(\theta; r^+_{3,m,i}) - \beta_m(r_{3,m,i}) && (190)\\ \Rightarrow A_m(\theta; r^-_{3,m,i}) - A_m(\theta; r^+_{3,m,i}) &= -(\alpha_1(\theta; r^-_{3,m,i}) - \alpha_1(\theta; r^+_{3,m,i})) && (191)\\ &= c\kappa_m\gamma_j(\theta; r_{3,m,i}) && (192)\\ &= c\kappa_m(\alpha_j(\theta; r_{3,m,i}) - \beta_j(r_{3,m,i})) && (193)\\ &= c\kappa_m A_j(\theta; r_{3,m,i}) && (194) \end{aligned}$$

Again note that at $t = r_{3,m,i}$, $A_j(\theta; t)$ is continuous since there are no changes in $\alpha_j(\theta; t)$ and $\beta_j(t)$ at that time. Hence $A_j(\theta; r^-_{3,m,i}) = A_j(\theta; r^+_{3,m,i}) = A_j(\theta; r_{3,m,i})$.
By a similar procedure, we can see that for $r_{4,m,i}(\theta)$ with $j > m, j = 1 \ldots, M$:

$$A_m(\theta; r^-_{4,m,i}) - A_m(\theta; r^+_{4,m,i}) = -c\kappa_m A_j(\theta; r^-_{4,m,i}) \quad (195)$$

Substituting Eq. (175), Eq. (188), Eq. (194) and Eq. (195) into Eq. (169) leads to:

$$\begin{aligned} \frac{dL_m(\theta; T)}{d\theta} &= \sum_{n \in \mathcal{F}_m} \Bigg[A_m(\theta; \varsigma^-_{m,n}) \frac{d\varsigma_{m,n}}{d\theta} - A_m(\theta; \tau^+_{m,n}) \frac{d\tau_{m,n}}{d\theta} \\ &\quad + \sum_{j=m+1}^{M} \sum_{\substack{i \in \Psi_{r,m,n} \\ j \to i}} c\kappa_m A_j(\theta; r_{3,m,i}) \frac{dr_{3,m,i}}{d\theta} \end{aligned}$$

$$
\begin{aligned}
&- \sum_{j=1}^{m-1} \sum_{\substack{i\in\Psi_{s,m,n} \\ j\to i}} \kappa_j A_j(\theta; s^-_{1,m,i}) \frac{ds_{1,m,i}}{d\theta} \\
&+ \sum_{l=m+1}^{M} \sum_{\substack{j=1 \\ l\to j}}^{m-1} \sum_{\substack{i\in\Upsilon_{s,m,n} \\ j\to i}} \frac{A_j(\theta; s^-_{2,m,i})}{A_l(\theta; s_{2,m,i})} A_l(\theta; s_{2,m,i}) \frac{ds_{2,m,i}}{d\theta} \\
&\left. - \sum_{l=m+1}^{M} \sum_{\substack{j=m+1 \\ l\to j}}^{M} \sum_{\substack{i\in\Upsilon_{r,m,n} \\ j\to i}} c\kappa_m \frac{A_j(\theta; r^-_{4,m,i})}{A_l(\theta; r_{4,m,i})} A_l(\theta; r_{4,m,i}) \frac{dr_{4,m,i}}{d\theta} \right] \qquad (196)
\end{aligned}
$$

Consider the term $A_m(\theta; \tau^+_{m,n})$.

$$
A_m(\theta; \tau^+_{m,n}) = \kappa_m A_m(\theta; \tau^-_{m,n}) \qquad (197)
$$

also

$$
A_m(\theta; \varsigma^-_{m,n}) \frac{d\varsigma_{m,n}}{d\theta} = \begin{cases} \frac{A_m(\theta;\varsigma^-_{m,n})}{A_j(\theta;\varsigma^-_{m,n})} \xi_j(\theta; \varsigma_{m,n}) & \text{if } n \in \Phi_m \text{ and if } Q_j \text{ at } \varsigma_{m,n} \\ & \text{caused the end of } B_{m,n} \\ 0 & \text{otherwise} \end{cases} \qquad (198)
$$

Using the definition in Eq. (121) and the continuity in $A_j(\theta; r_{3,m,i})$, Eq. (196) becomes

$$
\begin{aligned}
\frac{dL_m(\theta;T)}{d\theta} = \sum_{n\in\mathcal{F}_m} &\left[A_m(\theta; \varsigma^-_{m,n}) \frac{d\varsigma_{m,n}}{d\theta} - \kappa_m \xi_m(\theta; \tau_{m,n}) \right. \\
&\left. + \sum_{j=m+1}^{M} \sum_{\substack{i\in\Psi_{r,m,n} \\ j\to i}} c\kappa_m \xi_j(\theta; r_{3,m,i}) \right.
\end{aligned}
$$

$$
\begin{aligned}
&- \sum_{j=1}^{m-1} \sum_{\substack{i \in \Psi_{s,m,n} \\ j \to i}} \kappa_j \xi_j(\theta; s_{1,m,i}) \\
&+ \sum_{l=m+1}^{M} \sum_{\substack{j=1 \\ l \to j}}^{m-1} \sum_{\substack{i \in \Upsilon_{s,m,n} \\ j \to i}} \frac{A_j(\theta; s^-_{2,m,i})}{A_l(\theta; s_{2,m,i})} \xi_l(\theta; s_{2,m,i}) \\
&- \sum_{l=m+1}^{M} \sum_{\substack{j=m+1 \\ l \to j}}^{M} \sum_{\substack{i \in \Upsilon_{r,m,n} \\ j \to i}} c\kappa_m \frac{A_j(\theta; r^-_{4,m,i})}{A_l(\theta; r_{4,m,i})} \xi_l(\theta; r_{4,m,i})
\end{aligned}
\Bigg] \tag{199}
$$

5.9.2 Loss-feedback constant, c, as the control parameter, θ

The expression for the IPA loss-derivative with respect to the loss-feedback constant, c, follows the same format as that when the control parameter is b_m. However, the terms $\int_a^b \frac{dA_m(\theta;t)}{d\theta}\, dt$ in Eq. (168) can be non-zero due to the θ-dependence of $A_m(\theta;t)$ itself. In particular, when all the upstream queues are empty, i.e. $Q_i, i = 1, \ldots, m-1$, and because Q_m is full, then from Eq. (160) $\frac{dA_m(\theta;t)}{d\theta} = -\frac{\sigma(t)-\beta_m(t)+\sum \gamma_i(\theta;t)}{(1+c)^2}$. A closer examination of this expression shows that it is the total loss rate of the network (including lossy Q_m) during the time that the upstream queues $Q_i, i = 1, \ldots, m-1$ are empty, scaled by a factor of $\frac{1}{1+c}$. This is shown as follows:

The loss rate at Q_m:

$$
\begin{aligned}
\gamma_m(\theta;t) &= \frac{\sigma(t) + c\beta_m(t) - c\sum_{k=m+1}^{M} \gamma_k(\theta; s_{1,m,i})}{1+c} - \beta_m(t) && (200) \\
&= \frac{\sigma(t) - \beta_m(t) - c\sum_{k=m+1}^{M} \gamma_k(\theta; s_{1,m,i})}{1+c} && (201)
\end{aligned}
$$

The total loss rate for all the queues in the tandem

$$
\begin{aligned}
\sum_{k=m}^{M} \gamma_k(\theta; s_{1,m,i}) &= \frac{\sigma(t) - \beta_m(t) - c\sum_{k=m+1}^{M} \gamma_k(\theta; s_{1,m,i})}{1+c} + \sum_{k=m+1}^{M} \gamma_k(\theta; s_{1,m,i}) && (202) \\
&= \frac{\sigma(t) - \beta_m(t) + \sum_{k=m+1}^{M} \gamma_k(\theta; s_{1,m,i})}{1+c} && (203)
\end{aligned}
$$

Using the same definition of $L_{net}(\theta, a, b)$ which is total loss volume of all the queues in the tandem when $Q_i, i = 1, \ldots, m-1$ are empty during the period (a, b), we see that

$$\int_a^b \frac{dA_m(\theta;t)}{d\theta}\, dt \quad = \quad \frac{1}{1+c} L_{net}(\theta, a, b) \tag{204}$$

Hence:

$$\begin{aligned}
\frac{dL_m(\theta;T)}{d\theta} \quad = \quad & -\frac{1}{1+c} L_{net}(\theta, 0, T) + \sum_{n \in \mathcal{F}_m} \Bigg[A_m(\theta; \varsigma_{m,n}^-) \frac{d\varsigma_{m,n}}{d\theta} - \kappa_m \xi_m(\theta; \tau_{m,n}) \\
& + \sum_{j=m+1}^{M} \sum_{\substack{i \in \Psi_{r,m,n} \\ j \to i}} c\kappa_m \xi_j(\theta; r_{3,m,i}) \\
& - \sum_{j=1}^{m-1} \sum_{\substack{i \in \Psi_{s,m,n} \\ j \to i}} \kappa_j \xi_j(\theta; s_{1,m,i}) \\
& + \sum_{l=m+1}^{M} \sum_{\substack{j=1 \\ l \to j}}^{m-1} \sum_{\substack{i \in \Upsilon_{s,m,n} \\ j \to i}} \frac{A_j(\theta; s_{2,m,i}^-)}{A_l(\theta; s_{2,m,i})} \xi_l(\theta; s_{2,m,i}) \\
& - \sum_{l=m+1}^{M} \sum_{\substack{j=m+1 \\ l \to j}}^{M} \sum_{\substack{i \in \Upsilon_{r,m,n} \\ j \to i}} c\kappa_m \frac{A_j(\theta; r_{4,m,i}^-)}{A_l(\theta; r_{4,m,i})} \xi_l(\theta; r_{4,m,i}) \Bigg]
\end{aligned} \tag{205}$$

where

$$\kappa_j \quad = \quad \begin{cases} \frac{1}{1+c} & \text{if all } Q_i, i = 1, \ldots, j-1 \text{ are empty} \\ 1 & \text{otherwise} \end{cases} \tag{206}$$

and where

$$A_m(\theta;\varsigma_{m,n}^-)\frac{d\varsigma_{m,n}}{d\theta} = \begin{cases} \frac{A_m(\theta;\varsigma_{m,n}^-)}{A_j(\theta;\varsigma_{m,n}^-)}\xi_j(\theta;\varsigma_{m,n}) & \text{if } n \in \Phi_m \text{ and if } Q_j \text{ at } \varsigma_{m,n} \\ & \text{caused the end of } B_{m,n} \\ 0 & \text{otherwise} \end{cases} \tag{207}$$

5.10 IPA queue-workload derivative, $\frac{dQ(\theta,T)}{d\theta}$

The queue workload at node m, $Q_m(\theta;T)$, was given as

$$Q_m(\theta;T) \;=\; \int_0^T x_m(\theta;t)\,dt \tag{208}$$

Here, we take the approach of [78]. The period $[0,T]$ was partitioned into NBPs and BPs of which there were N_m BPs, so that Eq. (208) can be expressed as

$$Q_m(\theta;T) \;=\; \sum_{n=1}^{N_m}\left[\int_{\varsigma_{m,n-1}}^{\tau_{m,n}} x_m(\theta;t)\,dt + \int_{\tau_{m,n}}^{\varsigma_{m,n}} x_m(\theta;t)\,dt\right] \tag{209}$$

Now, $x_m(\theta;t)$ is continuous in t so that there are no jumps in $x_m(\theta;t)$. Taking derivatives of Eq. (209) with respect to θ:

$$\begin{aligned}\frac{dQ_m(\theta;T)}{d\theta} \;=\; \sum_{n=1}^{N_m}\Bigg[&\int_{\varsigma_{m,n-1}}^{\tau_{m,n}} \frac{dx_m(\theta;t)}{d\theta}\,dt + x_m(\theta;\tau_{m,n}^-)\frac{d\tau_{m,n}}{d\theta} - x_m(\theta;\varsigma_{m,n-1}^+)\frac{d\varsigma_{m,n-1}}{d\theta}\\ &+\int_{\tau_{m,n}}^{\varsigma_{m,n}} \frac{dx_m(\theta;t)}{d\theta}\,dt + x_m(\theta;\varsigma_{m,n}^-)\frac{d\varsigma_{m,n}}{d\theta} - x_m(\theta;\tau_{m,n}^+)\frac{d\tau_{m,n}}{d\theta}\Bigg]\end{aligned} \tag{210}$$

Due to continuity, there will be cancellation of terms. For example $x_m(\theta;\tau_{m,n}^-) = x_m(\theta;\tau_{m,n}^+)$, so that Eq. (210) reduces to

$$\begin{aligned}\frac{dQ_m(\theta;T)}{d\theta} \;=\; \sum_{n=1}^{N_m}&\left[\int_{\varsigma_{m,n-1}}^{\tau_{m,n}} \frac{dx_m(\theta;t)}{d\theta}\,dt + \int_{\tau_{m,n}}^{\varsigma_{m,n}} \frac{dx_m(\theta;t)}{d\theta}\,dt\right]\\ &+x_m(\theta;\varsigma_{m,N_m})\frac{d\varsigma_{m,N_m}}{d\theta} - x_m(\theta;\varsigma_{m,0})\frac{d\varsigma_{m,0}}{d\theta}\end{aligned} \tag{211}$$

From before, we assumed that $\varsigma_{m,N_m} = T$ and $\varsigma_{m,0} = 0$. Therefore, $\frac{d\varsigma_{m,N_m}}{d\theta} = 0$ and $\frac{d\varsigma_{m,0}}{d\theta} = 0$. As a result, Eq. (211) reduces to

$$\frac{dQ_m(\theta;T)}{d\theta} \;=\; \sum_{n=1}^{N_m}\left[\int_{\varsigma_{m,n-1}}^{\tau_{m,n}} \frac{dx_m(\theta;t)}{d\theta}\,dt + \int_{\tau_{m,n}}^{\varsigma_{m,n}} \frac{dx_m(\theta;t)}{d\theta}\,dt\right] \tag{212}$$

We need to determine expressions for $\frac{dx_m(\theta;t)}{d\theta}$. For $t \in B_{m,n}, n = 1,\ldots,N_m$:

$$x_m(\theta;t) \;=\; x_m(\theta;\tau_{m,n}) + \int_{\tau_{m,n}}^{t} \frac{dx_m(\theta;\nu)}{d\nu}\,d\nu \tag{213}$$

But for $t \in B_{m,n}$, $\frac{dx_m(\theta;t)}{dt} = 0$, therefore Eq. (213) becomes

$$x_m(\theta;t) \;=\; x_m(\theta;\tau_{m,n}) = \begin{cases} 0 & \text{if } B_{m,n} \text{ is an EP} \\ b_m & \text{if } B_{m,n} \text{ is an FP} \end{cases} \tag{214}$$

Taking derivatives of Eq. (214) with respect to θ:

$$\frac{dx_m(\theta;t)}{d\theta} \;=\; \frac{dx_m(\theta;\tau_{m,n})}{d\theta} = \begin{cases} 0 & \text{if } B_{m,n} \text{ is an EP} \\ 0 & \text{if } B_{m,n} \text{ is an FP but } \theta \text{ is not } b_m \\ 1 & \text{if } B_{m,n} \text{ is an FP and } \theta \equiv b_m \end{cases} \tag{215}$$

For $t \in NB_{m,n}, n = 1, \ldots, N_m$:

$$x_m(\theta;t) \;=\; x_m(\theta;\varsigma_{m,n-1}) + \int_{\varsigma_{m,n-1}}^{t} \frac{dx_m(\theta;\nu)}{d\nu}\, d\nu \tag{216}$$

But for $t \in NB_{m,n}$, $\frac{dx_m(\theta;t)}{dt} = A_m(\theta;t)$, therefore Eq. (216) becomes

$$x_m(\theta;t) \;=\; x_m(\theta;\varsigma_{m,n-1}) + \int_{\varsigma_{m,n-1}}^{t} A_m(\theta;\nu)\, d\nu \tag{217}$$

Taking derivatives of Eq. (217) with respect to θ:

$$\frac{dx_m(\theta;t)}{d\theta} \;=\; \frac{dx_m(\theta;\varsigma_{m,n-1})}{d\theta} + \frac{d}{d\theta}\int_{\varsigma_{m,n-1}}^{t} A_m(\theta;\nu)\, d\nu \tag{218}$$

Consider the first term in the *RHS* of Eq. (218). Since

$$x_m(\theta;\varsigma_{m,n-1}) \;=\; \begin{cases} 0 & \text{if } NB_{m,n} \text{ is an EE or EF} \\ b_m & \text{if } NB_{m,n} \text{ is an FE or FF} \end{cases} \tag{219}$$

then

$$\frac{dx_m(\theta;\varsigma_{m,n-1})}{d\theta} \;=\; \begin{cases} 0 & \text{if } NB_{m,n} \text{ is an EE or EF} \\ 0 & \text{if } NB_{m,n} \text{ is an FE or FF but } \theta \text{ is not } b_m \\ 1 & \text{if } NB_{m,n} \text{ is an FE or FF and } \theta \equiv b_m \end{cases} \tag{220}$$

Consider the second term in the *RHS* of Eq. (218), and again let $z_{m,n-1,i}$ be the i-th time-point in the interior of $NB_{m,n}$ at which there is a discontinuity in $A_m(\theta;t)$. Also, let $\Pi_{m,n-1}(t)$ be the number of such time-points in the interval $(\varsigma_{m,n-1}, t) \subset NB_{m,n}$.

$$\begin{aligned} \frac{d}{d\theta}\int_{\varsigma_{m,n-1}}^{t} A_m(\theta;\nu)\, d\nu \;=\; & \frac{d}{d\theta}\left[\int_{\varsigma_{m,n-1}}^{z_{m,n-1,1}} A_m(\theta;\nu)\, d\nu + \sum_{i=1}^{\Pi_{m,n-1}(t)-1}\left[\int_{z_{m,n-1,i}}^{z_{m,n-1,i+1}} A_m(\theta;\nu)\, d\nu\right]\right. \\ & \left. + \int_{z_{m,n-1,\Pi_{m,n-1}(t)}}^{t} A_m(\theta;\nu)\, d\nu\right] \end{aligned} \tag{221}$$

5.10.1 Buffer capacity, b_m, as the control parameter, θ

Following the same procedure as in Sections 5.8.1 and 5.9.1, Eq. (221) can be expressed as

$$\begin{aligned}\frac{d}{d\theta}\int_{\varsigma_{m,n-1}}^{t} A_m(\theta;\nu)\,d\nu &= -A_m(\theta;\varsigma^+_{m,n-1})\frac{d\varsigma_{m,n-1}}{d\theta} \\ &\quad + \sum_{i=1}^{\Pi_{m,n-1}(t)}\left[(A_m(\theta;z^-_{m,n-1,i}) - A_m(\theta;z^+_{m,n-1,i}))\frac{dz_{m,n-1,i}}{d\theta}\right]\end{aligned} \tag{222}$$

In its entirety, Eq. (218) will be:

$$\begin{aligned}\frac{dx_m(\theta;t)}{d\theta} &= \frac{dx_m(\theta;\varsigma_{m,n-1})}{d\theta} - A_m(\theta;\varsigma^+_{m,n-1})\frac{d\varsigma_{m,n-1}}{d\theta} \\ &\quad + \sum_{i=1}^{\Pi_{m,n-1}(t)}\left[(A_m(\theta;z^-_{m,n-1,i}) - A_m(\theta;z^+_{m,n-1,i}))\frac{dz_{m,n-1,i}}{d\theta}\right]\end{aligned} \tag{223}$$

The first integral term of Eq. (212) can be expressed as

$$\begin{aligned}\int_{\varsigma_{m,n-1}}^{\tau_{m,n}}\frac{dx_m(\theta;t)}{d\theta}\,dt &= \int_{\varsigma_{m,n-1}}^{z_{m,n-1,1}}\frac{dx_m(\theta;t)}{d\theta}\,dt + \sum_{i=1}^{K_{m,n-1}-1}\left[\int_{z_{m,n-1,i}}^{z_{m,n-1,i+1}}\frac{dx_m(\theta;t)}{d\theta}\,dt\right] \\ &\quad + \int_{z_{m,n-1,K_{m,n-1}}}^{\tau_{m,n}}\frac{dx_m(\theta;t)}{d\theta}\,dt\end{aligned} \tag{224}$$

Substituting Eq. (223) into Eq. (224), and after some cancellation of terms, leads to

$$\begin{aligned}\int_{\varsigma_{m,n-1}}^{\tau_{m,n}}\frac{dx_m(\theta;t)}{d\theta}\,dt &= \int_{\varsigma_{m,n-1}}^{z_{m,n-1,1}}\left(\frac{dx_m(\theta;\varsigma_{m,n-1})}{d\theta} - A_m(\theta;\varsigma^+_{m,n-1})\frac{d\varsigma_{m,n-1}}{d\theta}\right)dt \\ &\quad + \sum_{i=1}^{K_{m,n-1}-1}\left[\int_{z_{m,n-1,i}}^{z_{m,n-1,i+1}}\left(\frac{dx_m(\theta;\varsigma_{m,n-1})}{d\theta} - A_m(\theta;\varsigma^+_{m,n-1})\frac{d\varsigma_{m,n-1}}{d\theta}\right.\right. \\ &\quad \left.\left. + \sum_{k=1}^{i}\left((A_m(\theta;z^-_{m,n-1,k}) - A_m(\theta;z^+_{m,n-1,k}))\frac{dz_{m,n-1,k}}{d\theta}\right)\right)dt\right] \\ &\quad + \int_{z_{m,n-1,K_{m,n-1}}}^{\tau_{m,n}}\left(\frac{dx_m(\theta;\varsigma_{m,n-1})}{d\theta} - A_m(\theta;\varsigma^+_{m,n-1})\frac{d\varsigma_{m,n-1}}{d\theta}\right. \\ &\quad \left. + \sum_{k=1}^{K_{m,n-1}}\left((A_m(\theta;z^-_{m,n-1,k}) - A_m(\theta;z^+_{m,n-1,k}))\frac{dz_{m,n-1,k}}{d\theta}\right)\right)dt\end{aligned} \tag{225}$$

Further simplification of Eq. (225) results in:

$$\begin{aligned}
\int_{\varsigma_{m,n-1}}^{\tau_{m,n}} \frac{dx_m(\theta;t)}{d\theta}\,dt &= \left(\frac{dx_m(\theta;\varsigma_{m,n-1})}{d\theta} - A_m(\theta;\varsigma^+_{m,n-1})\frac{d\varsigma_{m,n-1}}{d\theta}\right)(z_{m,n-1,1} - \varsigma_{m,n-1}) \\
&+ \sum_{i=1}^{K_{m,n-1}-1}\Bigg[\Bigg(\frac{dx_m(\theta;\varsigma_{m,n-1})}{d\theta} - A_m(\theta;\varsigma^+_{m,n-1})\frac{d\varsigma_{m,n-1}}{d\theta} \\
&+ \sum_{k=1}^{i}\left((A_m(\theta;z^-_{m,n-1,k}) - A_m(\theta;z^+_{m,n-1,k}))\frac{dz_{m,n-1,k}}{d\theta}\right)\Bigg) \\
&\times(z_{m,n-1,i+1} - z_{m,n-1,i})\Bigg] \\
&+ \Bigg(\frac{dx_m(\theta;\varsigma_{m,n-1})}{d\theta} - A_m(\theta;\varsigma^+_{m,n-1})\frac{d\varsigma_{m,n-1}}{d\theta} \\
&+ \sum_{k=1}^{K_{m,n-1}}\left((A_m(\theta;z^-_{m,n-1,k}) - A_m(\theta;z^+_{m,n-1,k}))\frac{dz_{m,n-1,k}}{d\theta}\right)\Bigg) \times \\
&(\tau_{m,n} - z_{m,n-1,K_{m,n-1}}) \qquad (226) \\
&= \left(\frac{dx_m(\theta;\varsigma_{m,n-1})}{d\theta} - A_m(\theta;\varsigma^+_{m,n-1})\frac{d\varsigma_{m,n-1}}{d\theta}\right)(\tau_{m,n} - \varsigma_{m,n-1}) \\
&+ \sum_{k=1}^{K_{m,n-1}}\Bigg((A_m(\theta;z^-_{m,n-1,k}) - A_m(\theta;z^+_{m,n-1,k}))\frac{dz_{m,n-1,k}}{d\theta} \\
&\times(\tau_{m,n} - z_{m,n-1,k})) \qquad (227)
\end{aligned}$$

Returning to Eq. (212), the second integral term can be expressed as

$$\int_{\tau_{m,n}}^{\varsigma_{m,n}} \frac{dx_m(\theta;t)}{d\theta}\,dt = \frac{dx_m(\theta;\tau_{m,n})}{d\theta}(\varsigma_{m,n} - \tau_{m,n}) \qquad (228)$$

where $\frac{dx_m(\theta;\tau_{m,n})}{d\theta}$ is given in Eq. (215).

Hence, the expression for the IPA derivative for queue workload (Eq. (212))becomes:

$$\begin{aligned}
\frac{dQ_m(\theta;T)}{d\theta} &= \sum_{n=1}^{N_m}\Bigg[\left(\frac{dx_m(\theta;\varsigma_{m,n-1})}{d\theta} - A_m(\theta;\varsigma^+_{m,n-1})\frac{d\varsigma_{m,n-1}}{d\theta}\right)(\tau_{m,n} - \varsigma_{m,n-1}) \\
&+ \sum_{k=1}^{K_{m,n-1}}\left((A_m(\theta;z^-_{m,n-1,k}) - A_m(\theta;z^+_{m,n-1,k}))\frac{dz_{m,n-1,k}}{d\theta}(\tau_{m,n} - z_{m,n-1,k})\right) \\
&+ \frac{dx_m(\theta;\tau_{m,n})}{d\theta}(\varsigma_{m,n} - \tau_{m,n})\Bigg] \qquad (229)
\end{aligned}$$

Using previous definitions as in Section 5.8, Eq. (229) can be written as

$$\frac{dQ_m(\theta;T)}{d\theta} = \sum_{n=1}^{N_m}\left[\left(\frac{dx_m(\theta;\varsigma_{m,n-1})}{d\theta} - A_m(\theta;\varsigma^+_{m,n-1})\frac{d\varsigma_{m,n-1}}{d\theta}\right)(\tau_{m,n} - \varsigma_{m,n-1})\right.$$

$$
\begin{aligned}
&+ \sum_{j=m+1}^{M} \sum_{\substack{i \in \bar{\Psi}_{r,m,n} \\ j \to i}} c\xi_j(\theta; r_{3,m,i})(\tau_{m,n} - r_{3,m,i}) \\
&- \sum_{j=1}^{m-1} \sum_{\substack{i \in \bar{\Psi}_{s,m,n} \\ j \to i}} \xi_j(\theta; s_{1,m,i})(\tau_{m,n} - s_{1,m,i}) \\
&- \sum_{l=m+1}^{M} \sum_{j=1}^{m-1} \sum_{\substack{i \in \bar{\Upsilon}_{s,m,n} \\ l \to j \quad j \to i}} \frac{A_j(\theta; s^-_{2,m,i})}{A_l(\theta; s_{2,m,i})} \xi_l(\theta; s_{2,m,i})(\tau_{m,n} - s_{2,m,i}) \\
&+ \sum_{l=m+1}^{M} \sum_{j=m+1}^{M} \sum_{\substack{i \in \bar{\Upsilon}_{r,m,n} \\ l \to j \quad j \to i}} c\frac{A_j(\theta; r^-_{4,m,i})}{A_l(\theta; r_{4,m,i})} \xi_l(\theta; r_{4,m,i})(\tau_{m,n} - r_{4,m,i}) \\
&+ \frac{dx_m(\theta; \tau_{m,n})}{d\theta}(\varsigma_{m,n} - \tau_{m,n})\Big] \qquad (230)
\end{aligned}
$$

5.10.2 Loss-feedback constant, c, as the control parameter, θ

For $t \in B_{m,n}, n = 1, \ldots, N_m$, the expression for $\frac{dx_m(\theta;t)}{d\theta}$ when the loss-feedback constant, c, is the control parameter θ, is the same as that when the buffer capacity, b_m, is the control parameter. However, for $t \in NB_{m,n}, n = 1, \ldots, N_m$, the expression differs, since the expression for the term $\frac{d}{d\theta}\int_{\varsigma_{m,n-1}}^{t} A_m(\theta;\nu)\,d\nu$ is instead

$$
\begin{aligned}
\frac{d}{d\theta}\int_{\varsigma_{m,n-1}}^{t} A_m(\theta;\nu)\,d\nu &= \int_{\varsigma_{m,n-1}}^{t} \frac{dA_m(\theta;\nu)}{d\theta}\,d\nu - A_m(\theta;\varsigma^+_{m,n-1})\frac{d\varsigma_{m,n-1}}{d\theta} \qquad (231) \\
&\quad + \sum_{i=1}^{\Pi_{m,n-1}(t)} \left[(A_m(\theta; z^-_{m,n-1,i}) - A_m(\theta; z^+_{m,n-1,i}))\frac{dz_{m,n-1,i}}{d\theta}\right]
\end{aligned}
$$

By the analysis in Section 5.8.2, we find that

$$
\int_{\varsigma_{m,n-1}}^{t} \frac{dA_m(\theta;\nu)}{d\theta}\,d\nu = L_{net}(\theta, \varsigma_{m,n-1}, t) \qquad (232)
$$

where $L_{net}(\theta, \varsigma_{m,n-1}, t)$ is the total loss volume along the tandem during the times that all

upstream queues $Q_i, i = 1, \ldots, m-1$ are empty during the period $(\varsigma_{m,n-1}, t)$. Therefore Eq. (232) becomes

$$\begin{aligned}\frac{d}{d\theta}\int_{\varsigma_{m,n-1}}^{t} A_m(\theta;\nu)\,d\nu &= L_{net}(\theta,\varsigma_{m,n-1},t) - A_m(\theta;\varsigma^+_{m,n-1})\frac{d\varsigma_{m,n-1}}{d\theta} \\ &\quad + \sum_{i=1}^{\Pi_{m,n-1}(t)}\left[(A_m(\theta;z^-_{m,n-1,i}) - A_m(\theta;z^+_{m,n-1,i}))\frac{dz_{m,n-1,i}}{d\theta}\right]\end{aligned} \tag{233}$$

The *RHS* of Eq. (234) is similar to that when the buffer capacity, b_m, is the control parameter, θ, except for the term $L_{net}(\theta,\varsigma_{m,n-1},t)$. Following the same procedure as in the previous subsection (i.e. as when the buffer capacity, b_m, is the control parameter, we obtain the following expression for the IPA queue-workload derivative:

$$\begin{aligned}\frac{dQ_m(\theta;T)}{d\theta} &= \sum_{n=1}^{N_m}\Bigg[\int_{\varsigma_{m,n-1}}^{\tau_{m,n}} L_{net}(\theta,\varsigma_{m,n-1},t)dt - \left(A_m(\theta;\varsigma^+_{m,n-1})\frac{d\varsigma_{m,n-1}}{d\theta}\right)(\tau_{m,n}-\varsigma_{m,n-1}) \\ &\quad + \sum_{j=m+1}^{M}\sum_{\substack{i\in\bar{\Psi}_{r,m,n}\\ j\to i}} c\xi_j(\theta;r_{3,m,i})(\tau_{m,n}-r_{3,m,i}) \\ &\quad - \sum_{j=1}^{m-1}\sum_{\substack{i\in\bar{\Psi}_{s,m,n}\\ j\to i}} \xi_j(\theta;s_{1,m,i})(\tau_{m,n}-s_{1,m,i}) \\ &\quad - \sum_{l=m+1}^{M}\sum_{j=1}^{m-1}\sum_{\substack{i\in\bar{\Upsilon}_{s,m,n}\\ l\to j\quad j\to i}} \frac{A_j(\theta;s^-_{2,m,i})}{A_l(\theta;s_{2,m,i})}\xi_l(\theta;s_{2,m,i})(\tau_{m,n}-s_{2,m,i}) \\ &\quad + \sum_{l=m+1}^{M}\sum_{j=m+1}^{M}\sum_{\substack{i\in\bar{\Upsilon}_{r,m,n}\\ l\to j\quad j\to i}} c\frac{A_j(\theta;r^-_{4,m,i})}{A_l(\theta;r_{4,m,i})}\xi_l(\theta;r_{4,m,i})(\tau_{m,n}-r_{4,m,i})\Bigg]\end{aligned} \tag{234}$$

It should be noted that $\frac{dx_m(\theta;\varsigma_{m,n-1})}{d\theta}$ and $\frac{dx_m(\theta;\tau_{m,n})}{d\theta}$ are both always equal to zero.

5.11 Simulations with buffer capacity, b_m, as the control variable θ

The test configuration (Figure 3), parameters and procedures were the same as that for the single-stage case. (See Chapter 2, Section 2.3.) However, the service rate of the first queue, $\beta_1(t)$ is held constant at 10 Mbps, while that of the second queue, $\beta_2(t)$ is a random process, uniformly distributed between 6.75 Mbps and 11.25 Mbps. Its average service rate is 9 Mbps. The time interval between changes in the magnitude of $\beta_2(t)$ is a constant at 75 ms. Also, the second queue, Q_2, is not a simple droptail queue anymore but rather an IPA queue. The inflow rate, $\sigma(t)$ switched between either of two levels, 13.31 Mbps ("ON"-rate) and 3.32 Mbps ("OFF"-rate) every 100 ms or so. Its average was 8.32 Mbps. The feedback constant, c was 0.3333. The packet size was 554 bytes (512 bytes for the payload, and 42 bytes for the header). The time-horizon for the optimizations was, $T = 40$ second. For both the error analysis and optimization Matlab was used.

5.11.1 IPA derivative error-analysis

Because the two queue in the tandem calculate IPA derivatives, we needed to consider the case when the buffer capacity of Q_1, i.e., b_1, is the control parameter, and then when the buffer capacity of Q_2, i.e., b_2, is the control parameter. For each case we looked at these four derivatives: $\frac{dL_m}{db_1}$, $\frac{dQ_m}{db_1}$, $\frac{dL_m}{db_2}$ and $\frac{dQ_m}{db_2}$ $m = 1, 2$. When the control parameter is b_1, b_2 is held constant at 30 packets. When the control parameter is b_2, b_1 is held constant at 30 packets.

We ran simulations for a long period (i.e., $T = 240$ seconds) for three sets of four adjacent values of θ (in packets), namely $\{29, 30, 31, 32\}$, $\{50, 51, 52, 53\}$ and $\{99, 100, 101, 102\}$, and computed the IPA derivative in each case. Using the first-order approximation of the Taylor series expansion as in Eq. (30) we calculated the error. This process was repeated for three different seed values of the random number generator. The error values are presented in Table 6 to Table 9.

The magnitude of the percentage error in $\frac{dL_1}{db_1}$ never exceeded 0.32% while that of $\frac{dQ_1}{db_1}$ never exceeded 0.26%. For $\frac{dL_2}{db_1}$, the highest magnitude in percentage error was less than 1.4%,

whereas for $\frac{dQ_2}{db_1}$, the highest was 2.2%. For the most part, the magnitude of the percentage error in $\frac{dL_1}{db_2}$ fell between 1.0% and 2.0%, however, the highest was about 4.3%. For $\frac{dQ_1}{db_2}$, the highest was just over 4.0%. $\frac{dL_2}{db_2}$ percentage error (magnitude) did not exceed 3.0% (except in one instance for which it was 3.03%). The magnitude of the percentage error in $\frac{dQ_2}{db_2}$ did not exceed 1.5%.

5.11.2 IPA optimization

We carried out a two dimensional optimization where the control parameter vector θ consisted of $[b_1 b_2]^T$, the buffer limits of the two IPA nodes. The objective function to be optimized is:

$$J(\theta;T) = \frac{1}{T}E\left[w_{L_1}L_1(\theta;T) + w_{Q_1}Q_1(\theta;T) + w_{L_2}L_2(\theta;T) + w_{Q_2}Q_2(\theta;T)\right] \quad (235)$$

where the weights, $w_{L_1}, w_{Q_1}, w_{L_2}, w_{Q_2}$, which represent the tradeoff between loss and delay along the tandem, were taken as 14.4, 10.4, 40.0, 364, respectively.

The standard stochastic approximation algorithm was used, which is of the form

$$\theta_{n+1} = \theta_n - a_n \cdot H_n(\theta_n, \omega_n) + \Delta \;\; n = 0, 1, \ldots \quad (236)$$

Here $H_n(\theta_n, \omega_n)$ is an estimate of the derivative of $J(\theta;T)$ with respect to θ evaluated at $\theta = \theta_n$ over a sample path of ω_n. Specifically,

$$H_n(\theta_n, \omega_n) = \begin{bmatrix} g_1(\theta_n, \omega_n) \\ g_2(\theta_n, \omega_n) \end{bmatrix} = \begin{bmatrix} \frac{1}{T}\left(w_{L_1}\frac{dL_1(\theta_n;T)}{db_1} + w_{Q_1}\frac{dQ_1(\theta_n;T)}{db_1} + w_{L_2}\frac{dL_2(\theta_n;T)}{db_1} + w_{Q_2}\frac{dQ_2(\theta_n;T)}{db_1}\right) \\ \frac{1}{T}\left(w_{L_1}\frac{dL_1(\theta_n;T)}{db_2} + w_{Q_1}\frac{dQ_1(\theta_n;T)}{db_2} + w_{L_2}\frac{dL_2(\theta_n;T)}{db_2} + w_{Q_2}\frac{dQ_2(\theta_n;T)}{db_2}\right) \end{bmatrix} \quad (237)$$

where the derivatives $\frac{dL_m(\theta_n;T)}{db_j}, \frac{dQ_m(\theta_n;T)}{db_j}$ $m = 1, 2;\ j = 1, 2$ are the estimators derived using SFM-IPA for the loss volume and queue workload. Also a_n is the step-size taken to be:

$$a_n = [a_{n_1} a_{n_2}]^T \quad (238)$$

$$= [a_{0_1} n^{-\rho_1} a_{0_2} n^{-\rho_2}]^T \quad (239)$$

Table 6: IPA sensitivity analysis - the multi-stage case: $\theta \equiv b_1, m = 1$

	seed=1000					
b_1 (pkts)	$\frac{dL_1}{d\theta}$	L_1(bits)	$\epsilon_{L_1}(\%)$	$\frac{dQ_1}{d\theta}$	Q_1(bits-seconds)	$\epsilon_{Q_1}(\%)$
29	-904.4921	250184700	-0.13448	108.8912	14744820	-0.21048
30	-904.7969	246170600	-0.10324	108.429	15226410	-0.21378
31	-904.7969	242156400	0.26084	107.9632	15705940	-0.21796
32	-905.4281	238156800		107.4789	16183390	
50	-908.3225	165818100	0.27040	98.93487	24418980	-0.22563
51	-908.0229	161803300	-0.07809	98.50073	24856470	-0.24859
52	-907.5685	157775800	-0.03621	97.97101	25291940	-0.26888
53	-906.8185	153752000		97.45813	25724980	
99	0	0		0	41076460	
100	0	0		0	41076460	
101	0	0		0	41076460	
102	0	0		0	41076460	
	seed=5586					
b_1 (pkts)	$\frac{dL_1}{d\theta}$	L_1(bits)	$\epsilon_{L_1}(\%)$	$\frac{dQ_1}{d\theta}$	Q_1(bits-seconds)	$\epsilon_{Q_1}(\%)$
29	-903.3016	251824700	-0.07911	109.0809	14756430	-0.20200
30	-902.8016	247818100	-0.10455	108.6382	15238900	-0.20863
31	-902.8016	243812700	0.28033	108.1793	15719380	-0.20454
32	-902.9728	239822700		107.7441	16197850	
50	-905.431	167652700	0.26590	99.45215	24459790	-0.22731
51	-905.4694	163650500	-0.09369	98.99943	24899560	-0.23600
52	-905.4694	159633700	-0.08372	98.53695	25337290	-0.23030
53	-905.3095	155617300		98.08566	25773000	
99	0	0		0	41355650	
100	0	0		0	41355650	
101	0	0		0	41355650	
102	0	0		0	41355650	
	seed=9736					
b_1 (pkts)	$\frac{dL_1}{d\theta}$	L_1(bits)	$\epsilon_{L_1}(\%)$	$\frac{dQ_1}{d\theta}$	Q_1(bits-seconds)	$\epsilon_{Q_1}(\%)$
29	-903.7915	250791400	-0.10725	109.1631	14763100	-0.21100
30	-903.7915	246781500	-0.11224	108.7	15245890	-0.21139
31	-903.8916	242771400	0.31571	108.2354	15726630	-0.21666
32	-903.268	238778000		107.7911	16205290	
50	-906.2277	166629700	0.24403	99.34386	24469300	-0.25937
51	-906.5554	162623100	-0.10569	98.83937	24908450	-0.25478
52	-906.899	158601000	-0.06527	98.33406	25345390	-0.23096
53	-906.649	154579000		97.86503	25780200	
99	0	0		0	41246050	
100	0	0		0	41246050	
101	0	0		0	41246050	
102	0	0		0	41246050	

Table 7: IPA sensitivity analysis - the multi-stage case: $\theta \equiv b_1, m = 2$

	seed=1000					
b_1 (pkts)	$\frac{dL_2}{d\theta}$	L_2(bits)	$\epsilon_{L_2}(\%)$	$\frac{dQ_2}{d\theta}$	Q_2(bits-seconds)	$\epsilon_{Q_2}(\%)$
29	214.9769	111396300	0.42217	10.5	8244440	-0.62426
30	216.2976	112353100	0.27842	10.5	8290813	-0.01597
31	217.3794	113314400	0.95228	10.4	8337353	-0.53617
32	222.0414	114287000		10.3	8383388	
50	216.2309	131866700	0.13196	11.5	9270732	-0.17272
51	216.804	132826300	0.28356	11.4	9321561	-0.43828
52	219.992	133789900	-0.07226	11.3	9371745	0.22028
53	219.8106	134764200		11.3	9421842	
99	0	175933400		0	11029140	
100	0	175933400		0	11029140	
101	0	175933400		0	11029140	
102	0	175933400		0	11029140	
	seed=5586					
b_1 (pkts)	$\frac{dL_2}{d\theta}$	L_2(bits)	$\epsilon_{L_2}(\%)$	$\frac{dQ_2}{d\theta}$	Q_2(bits-seconds)	$\epsilon_{Q_2}(\%)$
29	198.8392	103023800	0.18663	9.1	7855048	-1.38443
30	199.9241	103906700	0.42168	8.86	7894827	-0.76076
31	202.138	104796500	1.34219	8.7	7933808	-2.37768
32	207.2605	105704400		8.53	7971436	
50	190.3724	121480300	0.41121	10.5	8787539	-1.15911
51	191.276	122327500	0.59745	10.4	8833717	-0.81099
52	192.517	123180300	-0.06274	10.3	8879282	-0.62210
53	192.3571	124033000		10.1	8924441	
99	0	162694400		0	10632340	
100	0	162694400		0	10632340	
101	0	162694400		0	10632340	
102	0	162694400		0	10632340	
	seed=9736					
b_1 (pkts)	$\frac{dL_2}{d\theta}$	L_2(bits)	$\epsilon_{L_2}(\%)$	$\frac{dQ_2}{d\theta}$	Q_2(bits-seconds)	$\epsilon_{Q_2}(\%)$
29	206.0295	108278400	-0.02439	10.3	7998558	-0.43627
30	206.4463	109191300	1.13993	10.2	8043922	-2.22982
31	210.641	110116700	0.56119	9.78	8088130	-1.61983
32	212.4397	111055500		9.33	8130769	
50	207.0459	127692900	0.69446	11.6	9038201	0.76834
51	209.6007	128616900	0.51130	11.6	9089986	-0.06076
52	212.5364	129550600	-0.12329	11.4	9141193	0.13769
53	212.2864	130491400		11.4	9191983	
99	0	171675600		0	10888220	
100	0	171675600		0	10888220	
101	0	171675600		0	10888220	
102	0	171675600		0	10888220	

Table 8: IPA sensitivity analysis - the multi-stage case: $\theta \equiv b_2, m = 1$

	seed=1000					
b_2 (pkts)	$\frac{dL_1}{d\theta}$	L_1(bits)	$\epsilon_{L_1}(\%)$	$\frac{dQ_1}{d\theta}$	Q_1(bits-seconds)	$\epsilon_{Q_1}(\%)$
29	119.1579	245650400	-1.49739	0.875063	15222600	-1.76055
30	115.5946	246170600	-1.44740	0.829007	15226410	-3.10708
31	112.0946	246675500	-2.43623	0.77919	15229970	-2.41413
32	108.0333	247160200		0.734853	15233340	
50	77.00346	254646300	-0.69718	0.369499	15272710	-1.07585
51	75.9815	254985200	-1.67780	0.383905	15274330	0.50156
52	73.72758	255316300	-0.41640	0.382257	15276040	-0.83594
53	72.47758	255641700		0.392453	15277720	
99	23.41246	264909700	-1.12180	0.17846	15333850	1.14632
100	22.66735	265012300	-2.84854	0.183367	15334650	0.90046
101	21.37655	265109900	-3.73743	0.185323	15335470	-3.81717
102	19.32584	265201100		0.181944	15336260	
	seed=5586					
b_2 (pkts)	$\frac{dL_1}{d\theta}$	L_1(bits)	$\epsilon_{L_1}(\%)$	$\frac{dQ_1}{d\theta}$	Q_1(bits-seconds)	$\epsilon_{Q_1}(\%)$
29	115.8352	247310400	-1.10670	0.879817	15235100	-2.54780
30	112.3268	247818100	-1.09121	0.83569	15238900	-3.61192
31	110.0735	248310500	-1.17778	0.78342	15242470	-3.80508
32	107.5735	248792600		0.733891	15245810	
50	74.88841	256010200	-1.41770	0.324998	15284410	-0.02709
51	73.07562	256337400	-1.68930	0.320314	15285850	-2.79156
52	71.32808	256655800	-1.21029	0.301891	15287230	-2.83877
53	69.0781	256968100		0.298167	15288530	
99	19.3602	265459400	-4.31727	0.162126	15331930	-3.97215
100	17.86021	265541500	-1.20830	0.154953	15332620	-6.80742
101	17.36021	265619700	-1.87216	0.136201	15333260	-0.60369
102	16.62387	265695200		0.157313	15333860	
	seed=9736					
b_2 (pkts)	$\frac{dL_1}{d\theta}$	L_1(bits)	$\epsilon_{L_1}(\%)$	$\frac{dQ_1}{d\theta}$	Q_1(bits-seconds)	$\epsilon_{Q_1}(\%)$
29	118.1265	246270300	-2.35641	0.877019	15242160	-4.03783
30	113.6209	246781500	-1.42341	0.822863	15245890	-3.20618
31	110.8031	247277900	-1.68595	0.765615	15249420	-0.97853
32	107.8523	247760700		0.728891	15252780	
50	79.42014	255198900	-0.99152	0.222783	15284020	0.26616
51	77.61228	255547400	-1.15636	0.221849	15285010	-1.34604
52	75.84708	255887400	-0.87884	0.223853	15285980	-1.22106
53	74.59366	256220600		0.219087	15286960	
99	20.72621	265167900	-0.71693	0.16466	15332810	-1.33926
100	20.71474	265259100	-2.18688	0.156003	15333530	-3.09591
101	19.46475	265348900	-0.65815	0.156648	15334200	-3.49492
102	19.21475	265434600		0.147271	15334870	

Table 9: IPA sensitivity analysis - the multi-stage case: $\theta \equiv b_2, m = 2$

	seed=1000					
b_2 (pkts)	$\frac{dL_2}{d\theta}$	L_2(bits)	$\epsilon_{L_2}(\%)$	$\frac{dQ_2}{d\theta}$	Q_2(bits-seconds)	$\epsilon_{Q_2}(\%)$
29	-539.679	114727300	0.73821	56.27248	8042902	-0.59689
30	-530.086	112353100	0.89134	55.75114	8290813	-0.39583
31	-519.463	110024700	1.75746	55.24768	8536924	-0.27556
32	-505.226	107762900		54.99673	8781107	
50	-377.207	72266400	1.01944	46.73769	12840570	-1.21243
51	-370.653	70611660	0.93403	45.50486	13045200	-0.45495
52	-363.316	68984270	0.67541	44.99512	13245960	-0.36525
53	-359.351	67384930		44.78569	13444650	
99	-128.277	19517820	0.16560	22.54155	20202950	0.69652
100	-126.651	18950240	1.58684	22.48901	20303550	-1.16512
101	-122.42	18397830	2.89121	22.0853	20402060	-2.74008
102	-113.392	17870950		21.00776	20497260	
	seed=5586					
b_2 (pkts)	$\frac{dL_2}{d\theta}$	L_2(bits)	$\epsilon_{L_2}(\%)$	$\frac{dQ_2}{d\theta}$	Q_2(bits-seconds)	$\epsilon_{Q_2}(\%)$
29	-528.31	106222500	1.09626	52.12498	7665483	-0.72458
30	-515.934	103906700	1.19032	51.50016	7894827	-0.68027
31	-505.182	101647300	0.53810	50.81053	8121523	-0.54321
32	-500.01	99420380		50.26328	8345492	
50	-363.142	65310110	0.69693	41.96572	12036680	-0.38823
51	-358.506	63711880	1.99046	41.64537	12221950	-0.99813
52	-346.552	62154610	1.11590	40.75325	12404680	-0.55831
53	-337.638	60635830		40.25876	12584290	
99	-117.742	17224950	3.02628	20.82697	18658320	-1.43560
100	-110.89	16718910	1.63702	20.14481	18749300	-0.88685
101	-106.444	16235490	2.46289	19.68996	18837790	-1.12977
102	-101.711	15775350		19.2026	18924070	
	seed=9736					
b_2 (pkts)	$\frac{dL_2}{d\theta}$	L_2(bits)	$\epsilon_{L_2}(\%)$	$\frac{dQ_2}{d\theta}$	Q_2(bits-seconds)	$\epsilon_{Q_2}(\%)$
29	-535.59	111521800	1.82139	52.95748	7810611	-0.59502
30	-519.148	109191300	0.82856	52.23708	8043922	-0.07116
31	-511.843	106909500	0.72676	52.1392	8275272	-0.32194
32	-503.06	104657500		51.68238	8505609	
50	-372.694	69862580	0.53756	42.90681	12277260	-0.23295
51	-367.25	68219680	1.48757	42.63646	12466980	-0.35181
52	-357.648	66616240	1.17767	42.19385	12655280	-0.48830
53	-350.629	65049810		41.90968	12841370	
99	-124.649	18719110	1.05932	22.34495	19303810	-0.30578
100	-123.137	18172520	2.14995	22.128	19402540	-1.02099
101	-118.35	17638510	0.96613	21.81028	19499610	-0.63433
102	-115.228	17119050		21.47247	19595660	

where $a_{0_1} = 1$, $a_{0_2} = 1$, $\rho_1 = 0.6$ and $\rho_2 = 0.6$. Each component of Δ is calculated as in Section 2.3.2.

The time-horizon, T, for each iteration was $T = 40$ second. The length of each optimization was 200 iterations. Three sets of optimizations were performed, having different initial values of θ, i.e., $\theta_0 = [10, 30]$, $\theta_0 = [30, 50]$, and $\theta_0 = [50, 80]$. For each set of optimizations, two types were performed. For the first type, each iteration uses the same seed for the random number generator, (i.e., $\omega_n = \omega$ for $n = 1, 2, \ldots, 200$). In this case, if the IPA derivatives were accurate themselves, the algorithm will converge to θ^* where θ^* is a minima of the cost function $J(\theta; T)$ evaluated using the same sample-path ω. For the second type, each iteration was independent of the others, i.e. a different random seed for the random generator.

The contour map of the cost function is shown in Figure 19a. The minima occurs at $\theta = [27, 35]$, so that for the deterministic optimization, covergence should be to $\theta = [27, 35]$. For the stochastic optimization (i.e., ω_n changes with each iteration), the convergence may occur in the vicinity of $\theta^* = [27, 35]$. This trajectory can be seen to be the case in Figure 19 for all three sets of initial values.

5.12 Simulations with loss-feedback constant, c, as the control variable θ

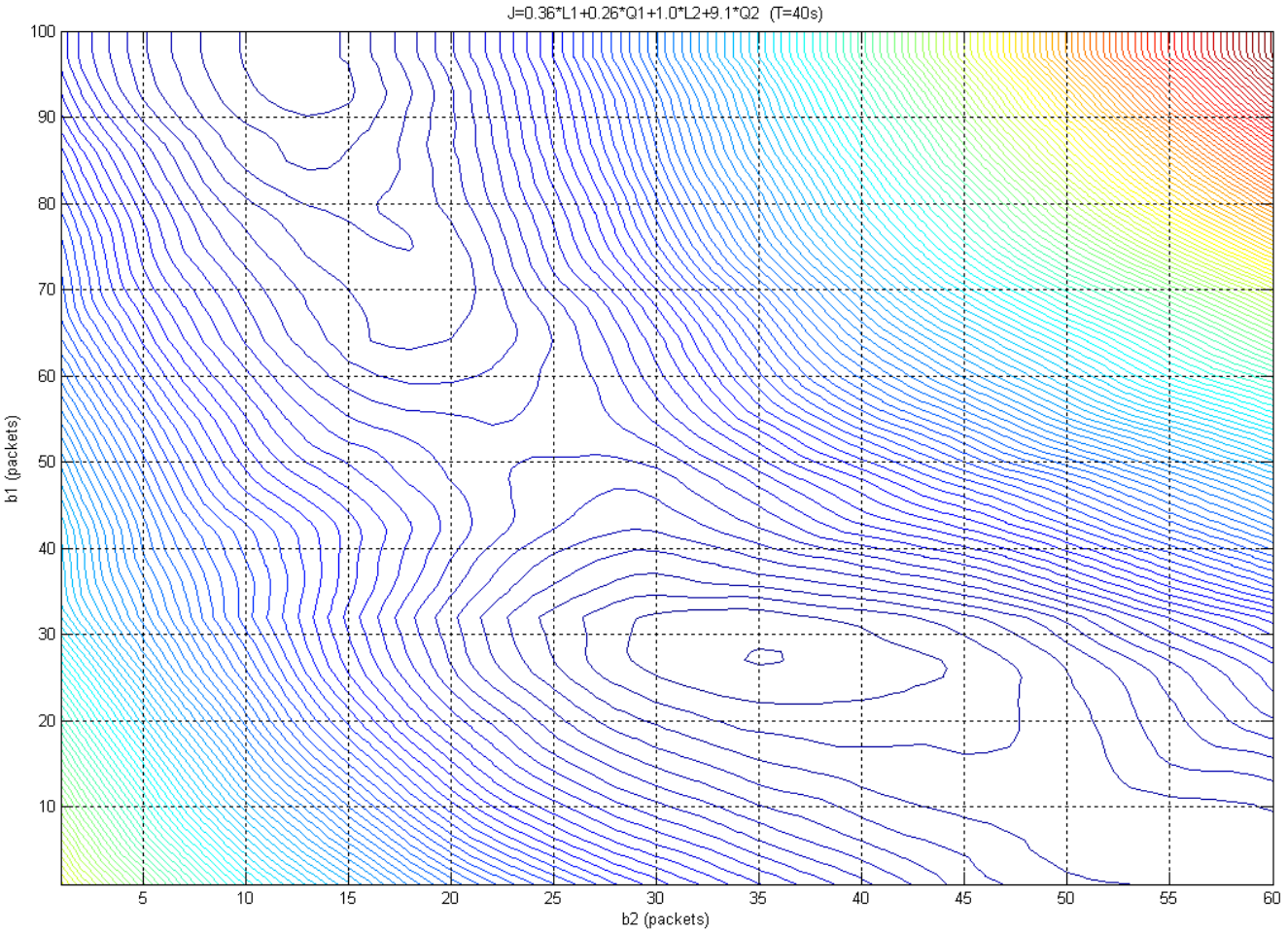
The same test configuration was used as in Figure 3. For both the error analysis and optimization Matlab was used.

5.12.1 IPA derivative error-analysis

For the error analysis, the buffer capacity of Q_1, i.e., b_1, was held constant at 30 packets, and that of Q_2, i.e., b_2, was also held at 30 packets. We performed the alternative approach to analysis by using three sets of four adjacent values of the control variable, c, and three different seeds of the random number generator. See Table 10 and Table 11

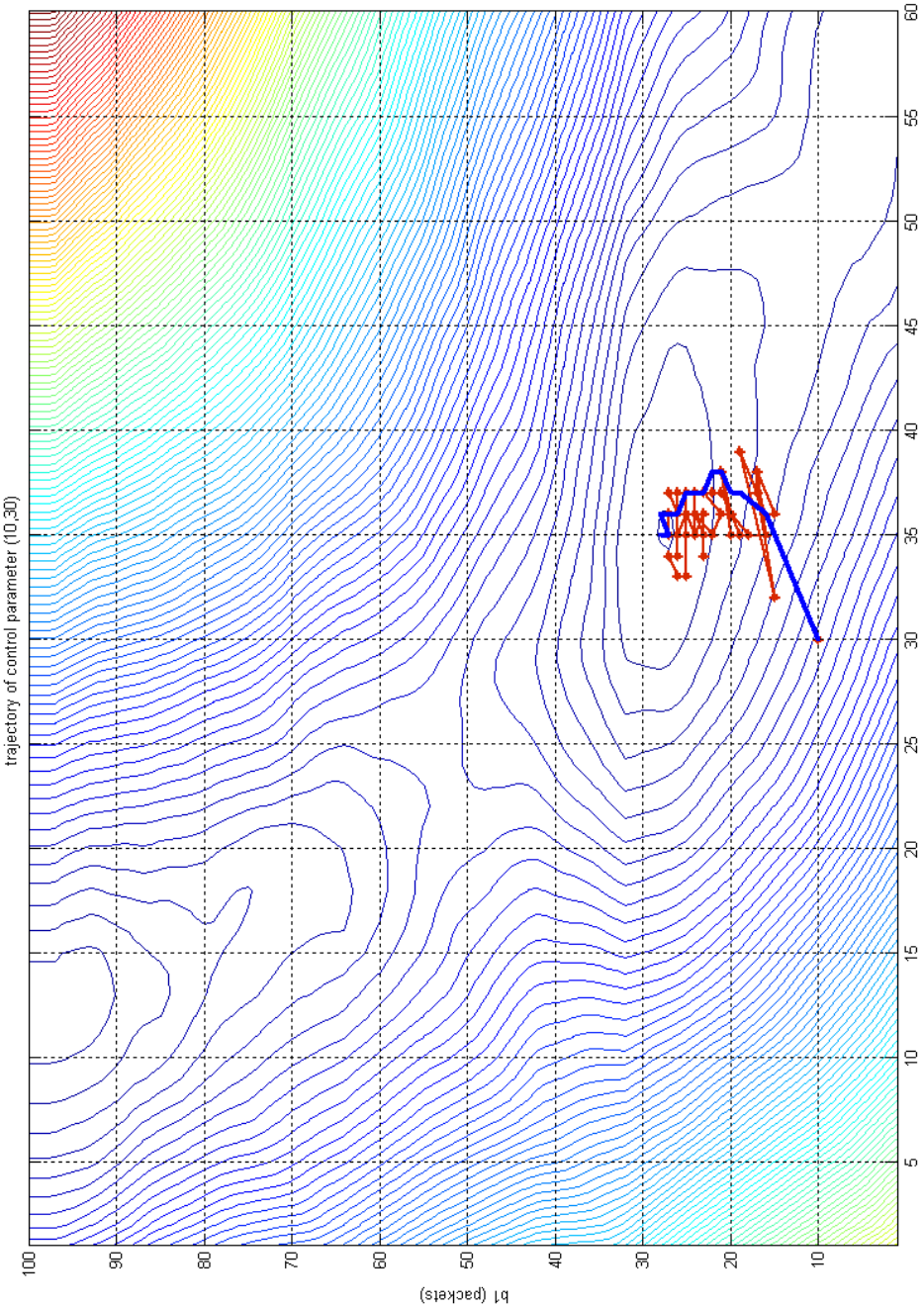
5.12.2 IPA Optimization

The optimizations were carried out using the same procedure as in Section 2.4.2, however b_1 and b_2 were both held constant at 30 packets. Also $a_0 = 0.015$. The cost function was take as



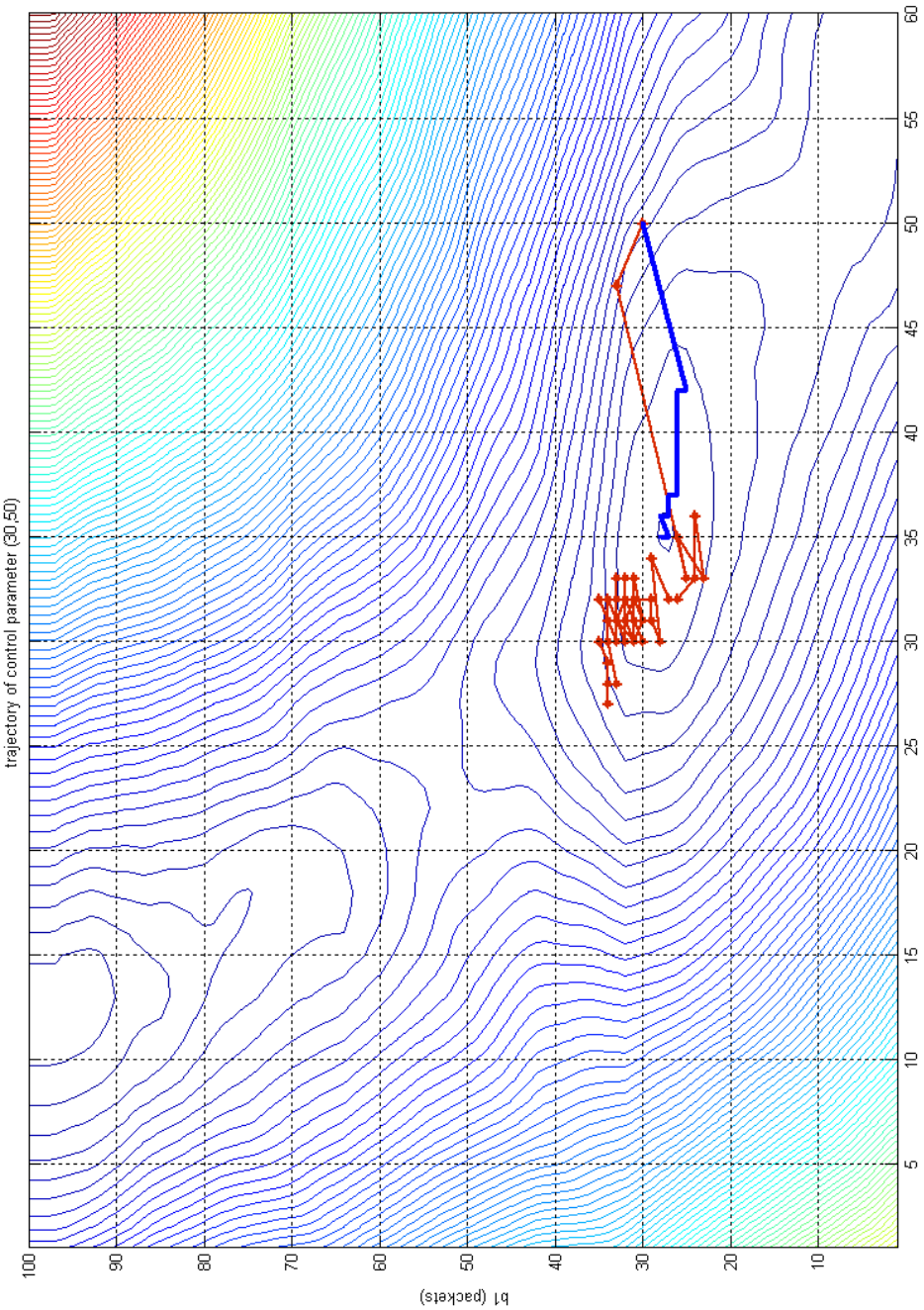
(a) Cost (seed = 1000)

Figure 19: IPA optimization with respect to b_1 and b_2: 2-dimensional trajectory



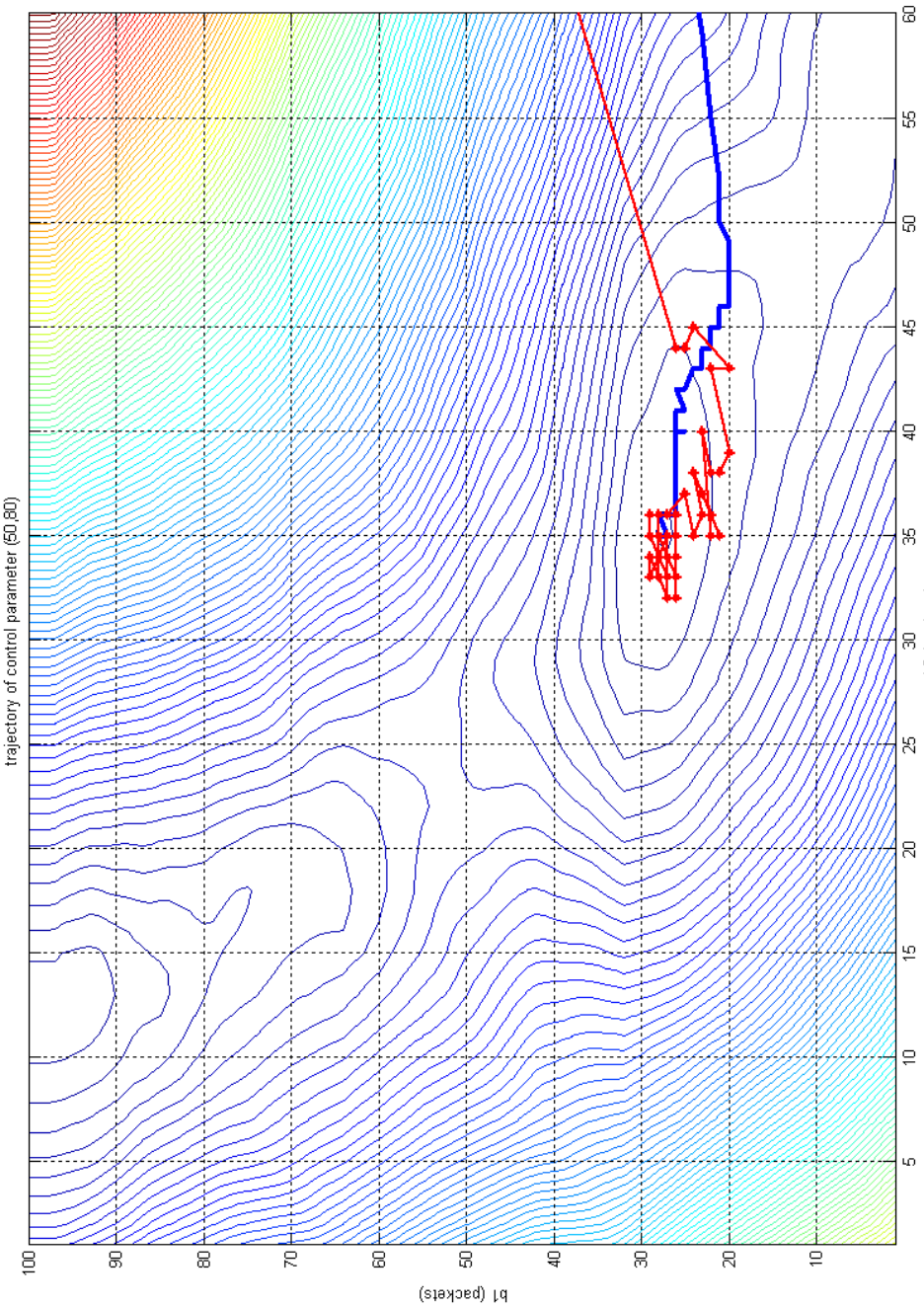
(b) Initial $(b_1, b_2) = (10, 30)$

Figure 19: IPA optimization with respect to b_1 and b_2: 2-dimensional trajectory



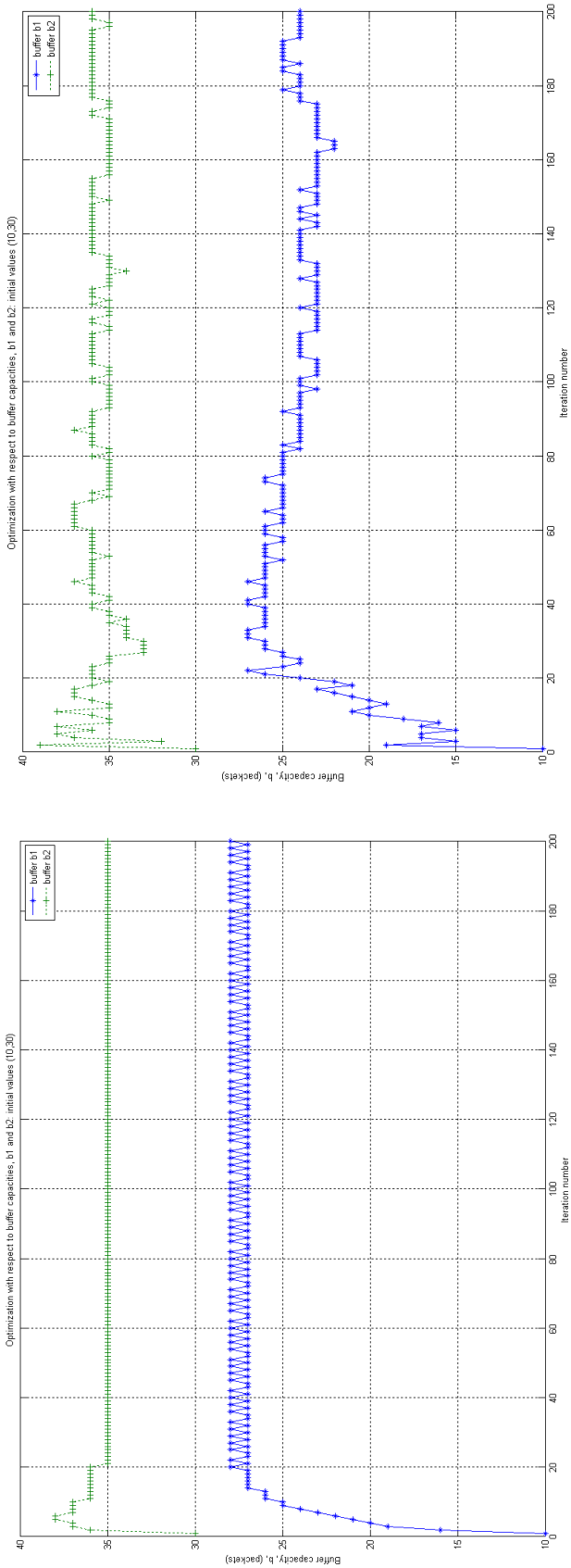
(c) Initial $(b_1, b_2) = (30, 50)$

Figure 19: IPA optimization with respect to b_1 and b_2: 2-dimensional trajectory



(d) Initial $(b_1, b_2) = (50, 80)$

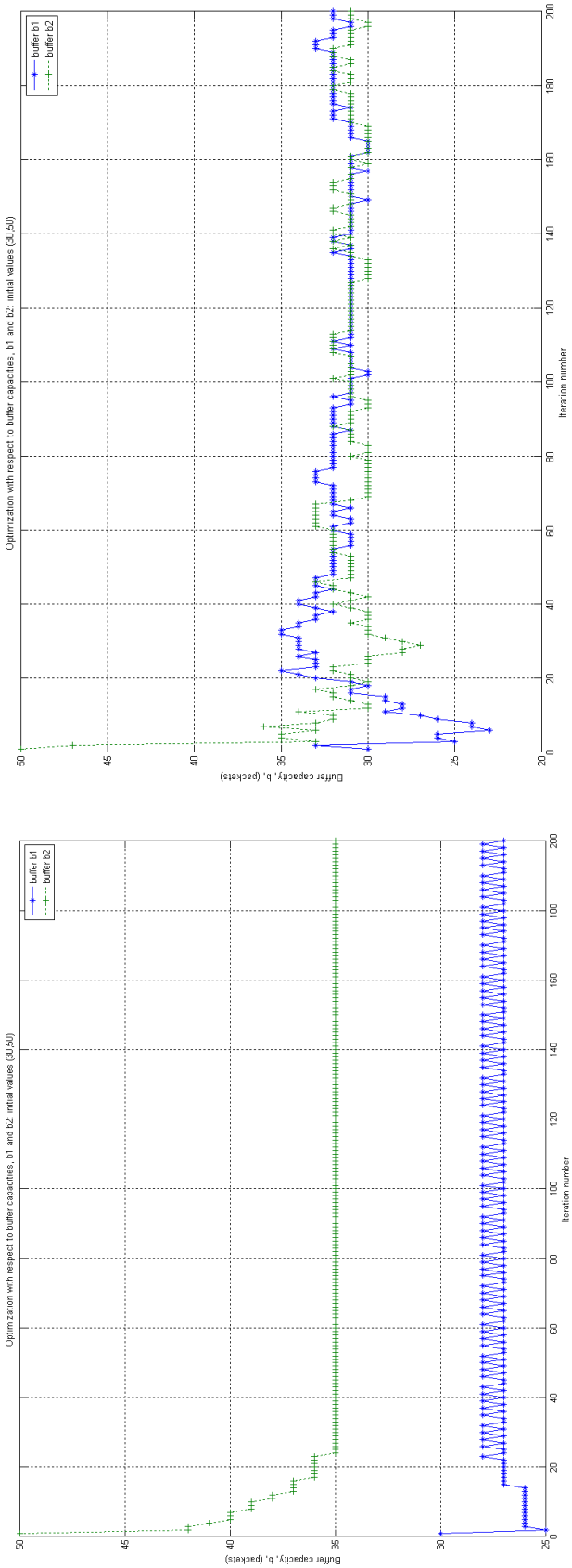
Figure 19: IPA optimization with respect to b_1 and b_2: 2-dimensional trajectory



(a) Initial $(b_1, b_2) = (10, 30)$ - deterministic

(b) Initial $(b_1, b_2) = (10, 30)$ - random

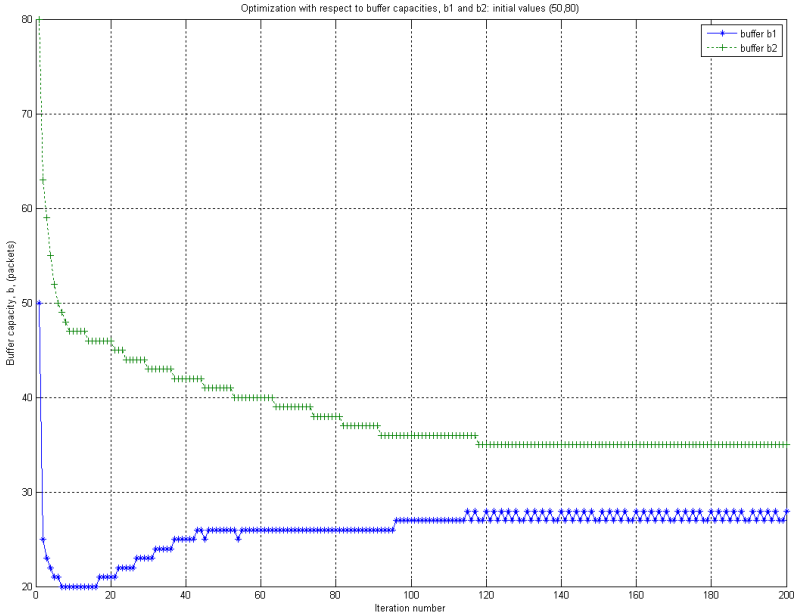
Figure 20: IPA optimization with respect to b_1 and b_2: convergence



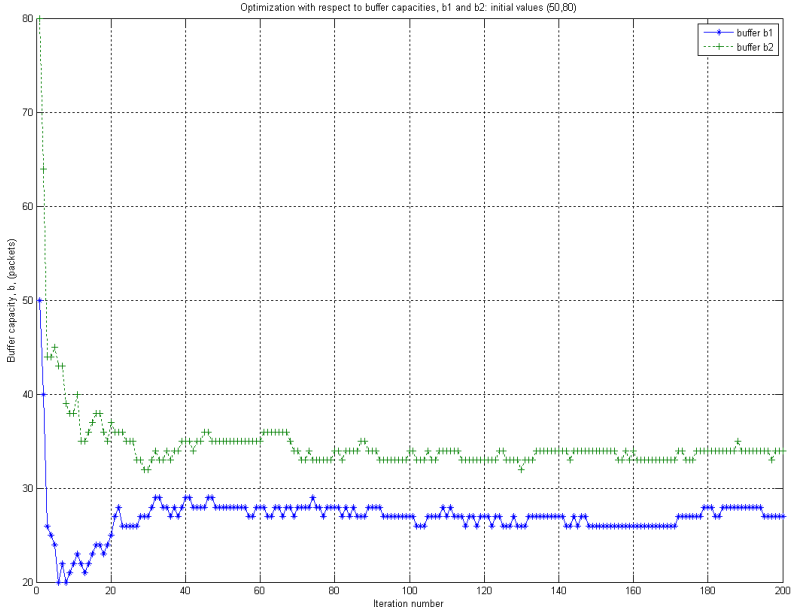
(c) Initial $(b_1, b_2) = (30, 50)$ - deterministic

(d) Initial $(b_1, b_2) = (30, 50)$ - random

Figure 20: IPA optimization with respect to b_1 and b_2: convergence

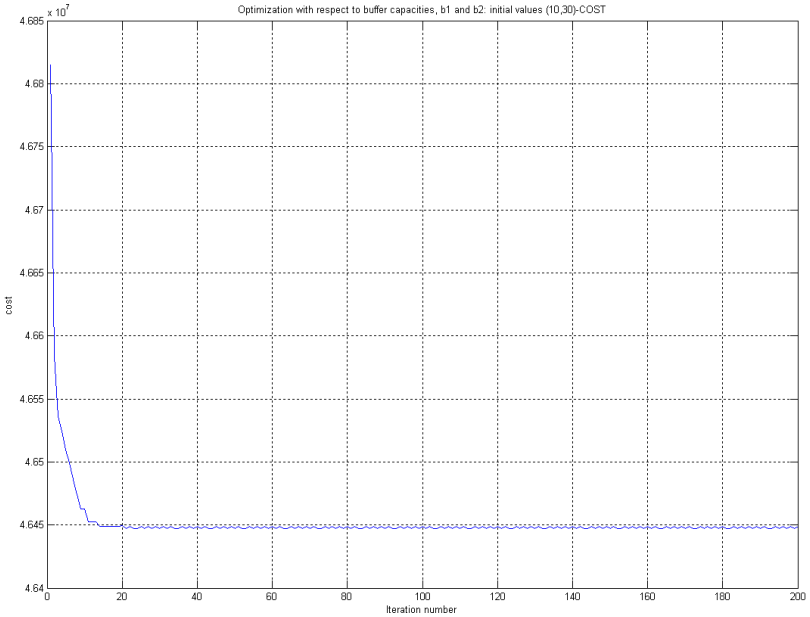


(e) Initial $(b_1, b_2) = (50, 80)$ - deterministic

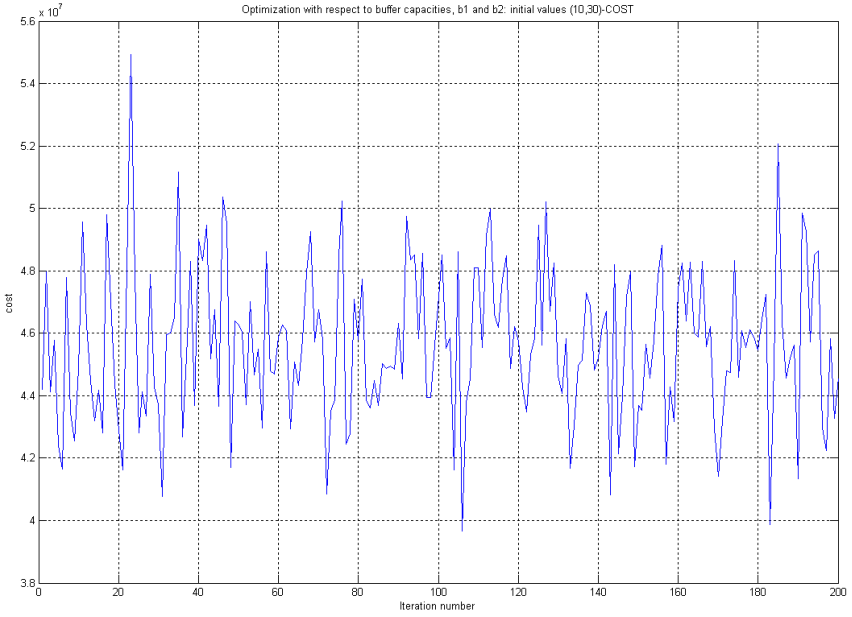


(f) Initial $(b_1, b_2) = (50, 80)$ - random

Figure 20: IPA optimization with respect to b_1 and b_2: convergence

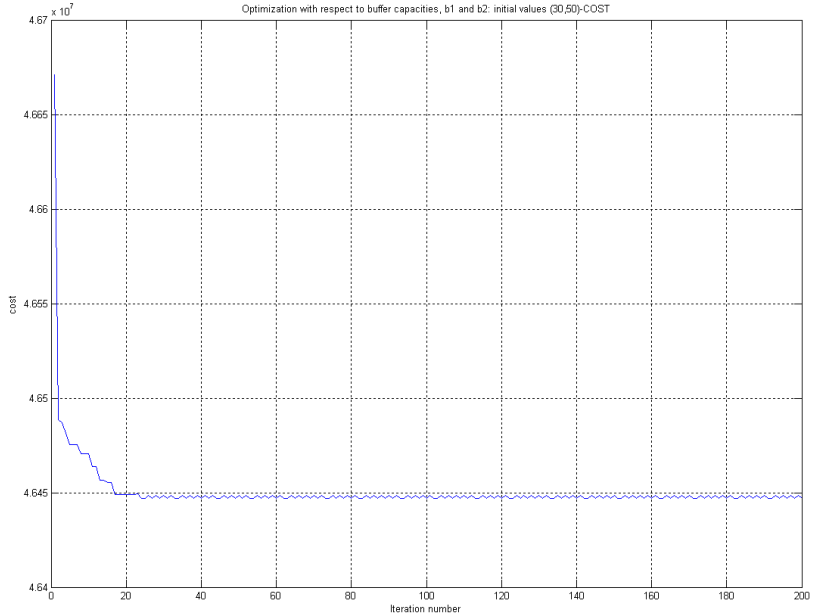


(a) Initial $(b_1, b_2) = (10, 30)$ - deterministic

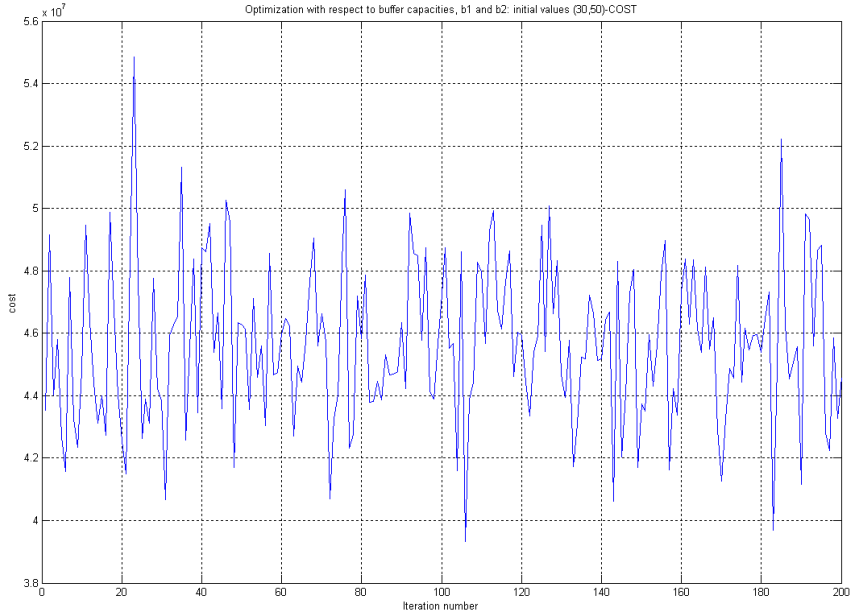


(b) Initial $(b_1, b_2) = (10, 30)$ - random

Figure 21: IPA optimization with respect to b_1 and b_2: cost progression

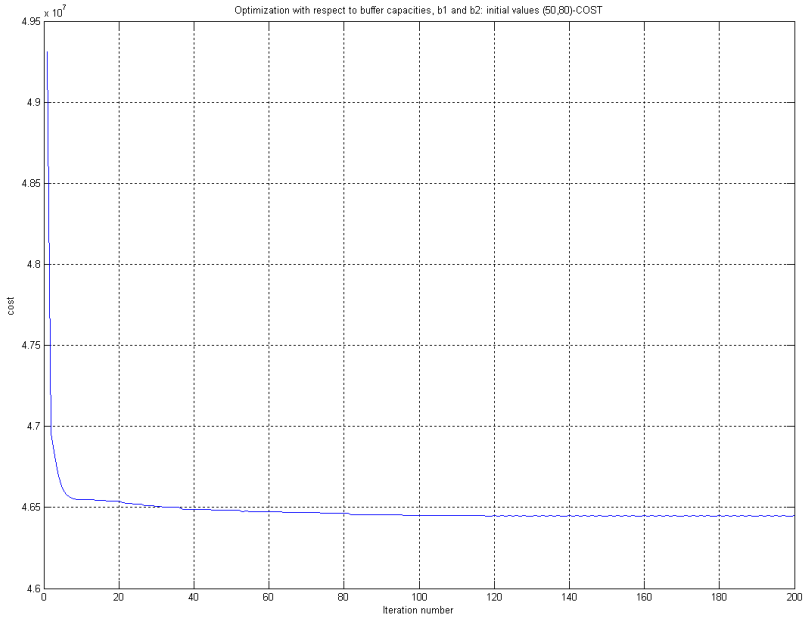


(c) Initial $(b_1, b_2) = (30, 50)$ - deterministic

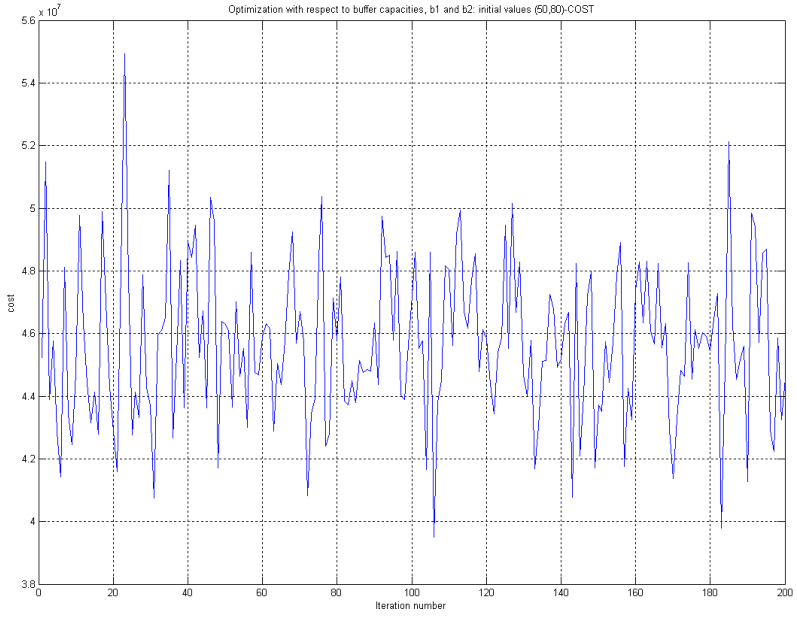


(d) Initial $(b_1, b_2) = (30, 50)$ - random

Figure 21: IPA optimization with respect to b_1 and b_2: cost progression



(e) Initial $(b_1, b_2) = (50, 80)$ - deterministic



(f) Initial $(b_1, b_2) = (50, 80)$ - random

Figure 21: IPA optimization with respect to b_1 and b_2: cost progression

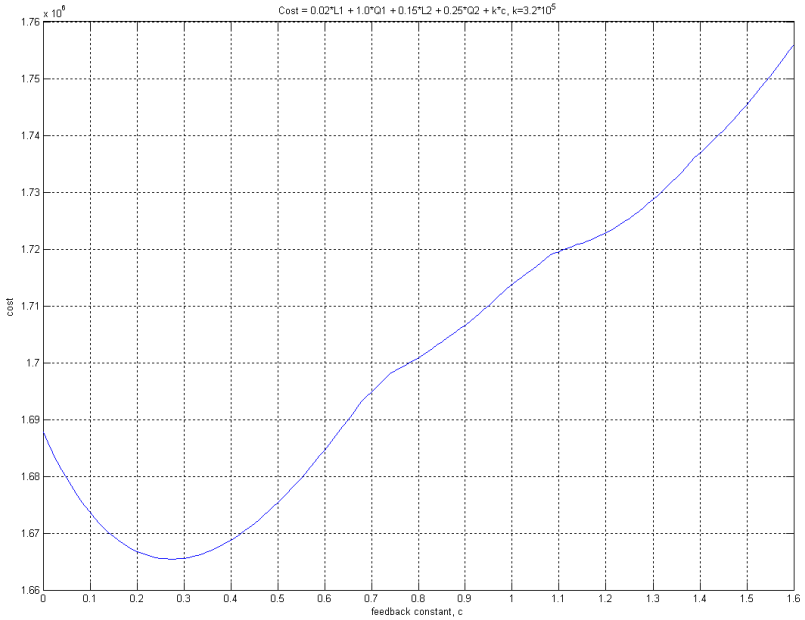
Table 10: IPA sensitivity analysis - the multi-stage case: $\theta \equiv c, m = 1$

	seed=1000					
c	$\frac{dL_1}{d\theta}$	L_1(bits)	$\epsilon_{L_1}(\%)$	$\frac{dQ_1}{d\theta}$	Q_1(bits-seconds)	$\epsilon_{Q_1}(\%)$
0	-445347299.5	357451368.6	1.01877	-421550	15362969.75	-1.67806
0.01	-436560225.1	353043266.1	0.99017	-417613	15358683.51	-1.80943
0.02	-428029709.5	348720890.8	0.98413	-413898	15354431.82	-1.93686
0.03	-419746706.7	344482717.3		-410373	15350212.68	
0.7	-147021455.5	174640358.7	0.90016	-1185026	14960842.95	-17.97377
0.71	-144333336.8	173183378.5	0.91331	-1230079	14946862.75	-17.82202
0.72	-141426002.9	171753227.2	0.75745	-1264860	14932369.71	-16.23818
0.73	-139257162	170349679.5		-1283338	14917667.22	
1.57	-45187149.78	104400161.9	0.49634	-1056662	13393232.36	-15.08790
1.58	-44764165.46	103950533.2	0.41525	-1046338	13381071.46	-13.75036
1.59	-44417861.63	103504750.4	0.59338	-1033117	13369169.33	-14.17924
1.6	-43941119.27	103063207.4		-1024978	13357373.28	
	seed=5586					
c	$\frac{dL_1}{d\theta}$	L_1(bits)	$\epsilon_{L_1}(\%)$	$\frac{dQ_1}{d\theta}$	Q_1(bits-seconds)	$\epsilon_{Q_1}(\%)$
0	-438690181.8	357451368.6	1.02039	-384614	15362969.75	-1.86854
0.01	-430039085.5	353109230.3	0.99139	-381003	15359051.75	-2.01513
0.02	-421640434.4	348851472.8	0.97732	-377562	15355164.94	-2.16737
0.03	-413485333.8	344676276.4		-374301	15351307.49	
0.7	-144895306.4	177218180.2	1.04234	-1126853	14995857.11	-13.31948
0.71	-141871402.3	175784330.2	0.73636	-1187833	14983087.67	-12.04605
0.72	-139485168.2	174376063	1.29261	-1227635	14969778.47	-14.42848
0.73	-136245230.5	172999241.4		-1289093	14955730.83	
1.57	-45529243.82	106702188.3	0.43762	-1085208	13535129.71	-15.79919
1.58	-45148967.25	106248888.3	0.50766	-1073546	13522563.09	-14.93390
1.59	-44748401.28	105799690.7	0.43323	-1052191	13510224.4	-15.46487
1.6	-44404718.15	105354145.3		-1038060	13498075.29	
	seed=9736					
c (pkts)	$\frac{dL_1}{d\theta}$	L_1(bits)	$\epsilon_{L_1}(\%)$	$\frac{dQ_1}{d\theta}$	Q_1(bits-seconds)	$\epsilon_{Q_1}(\%)$
0	-442806722.5	357451368.6	1.01573	-385655	15362969.75	-0.71805
0.01	-434076622.8	353068278.4	0.99235	-381162	15359085.51	-0.80617
0.02	-425601493.4	348770587.6	0.97894	-376849	15355243.17	-0.89741
0.03	-417371941.9	344556236.3		-372726	15351440.86	
0.7	-144901200.6	175565518	1.13075	-1201355	15025464.82	-15.95280
0.71	-142111773.3	174132890.7	0.65157	-1262115	15011534.77	-12.67371
0.72	-140181715.4	172721032.6	0.95735	-1279120	14997314.05	-15.35219
0.73	-137345954.9	171332635.7		-1354953	14982559.12	
1.57	-45588996.65	106088007	0.51572	-1057237	13434567.23	-12.85867
1.58	-45141295.74	105634468.1	0.49995	-1045921	13422635.4	-13.59710
1.59	-44641557.2	105185312	0.68856	-1039510	13410754.04	-14.60062
1.6	-44104347.51	104741970.2		-1028034	13398841.18	

Table 11: IPA sensitivity analysis - the multi-stage case: $\theta \equiv c, m = 2$

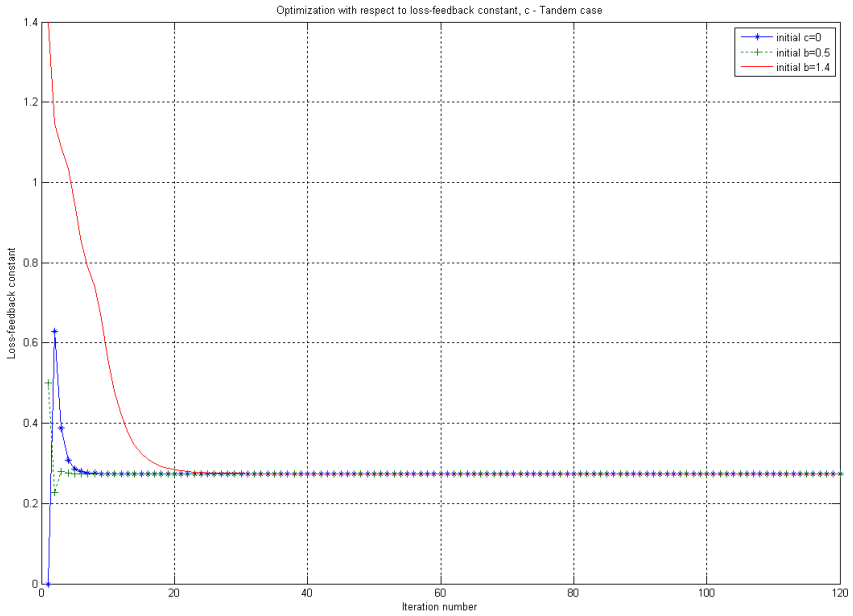
	seed=1000					
c	$\frac{dL_2}{d\theta}$	L_2(bits)	$\epsilon_{L_2}(\%)$	$\frac{dQ_2}{d\theta}$	Q_2(bits-seconds)	$\epsilon_{Q_2}(\%)$
0	-26010893.89	119086227.6	0.92293	-267032	8367721.446	0.61783
0.01	-25566598.26	118828519.3	1.07353	-264603	8365067.62	0.60236
0.02	-25134438.16	118575598	0.79505	-262206	8362437.529	0.39788
0.03	-24713110.75	118326251.9		-259836	8359825.901	
0.7	-15120356.91	107411791.4	-2.42465	-222173	8221068.359	0.28990
0.71	-15701641.54	107256921.7	-1.25591	-222196	8218853.066	0.73453
0.72	-16380109.22	107097933.3	-0.96395	-221454	8216647.431	1.40194
0.73	-16706471.22	106932553.3		-220910	8214463.94	
1.57	-25615661.16	86623741	0.05355	-228648	7957464.806	-0.77590
1.58	-25567464.96	86367721.56	-0.36153	-230233	7955160.581	2.53448
1.59	-25568393.83	86111122.57	0.24295	-222661	7952916.6	-0.48849
1.6	-25566416.56	85856059.8		-222731	7950679.113	
	seed=5586					
c	$\frac{dL_2}{d\theta}$	L_2(bits)	$\epsilon_{L_2}(\%)$	$\frac{dQ_2}{d\theta}$	Q_2(bits-seconds)	$\epsilon_{Q_2}(\%)$
0	-23247962.56	109952505.9	1.08193	-230998	7961373.202	0.81059
0.01	-22858432.8	109722541.5	0.79914	-228852	7959081.945	0.43491
0.02	-22478651.76	109495783.9	0.85031	-226733	7956803.375	0.46312
0.03	-22108599.54	109272908.7		-224646	7954546.541	
0.7	-14878600.64	99446441.53	-2.70392	-172337	7837370.356	-5.45823
0.71	-15766978.71	99293632.47	-0.44006	-197239	7835552.925	-0.51460
0.72	-16090890.54	99135268.85	-3.49096	-206569	7833570.381	-2.08893
0.73	-16811049.67	98968742.68		-212768	7831461.536	
1.57	-23157703.82	80496749.15	0.35072	-241653	7570334.515	0.05407
1.58	-23067745.39	80265984.3	0.08610	-236598	7567919.292	0.59056
1.59	-22978141.36	80035505.45	0.34949	-233665	7565567.279	0.36463
1.6	-22819110.76	79806527.11		-232181	7563239.151	
	seed=9736					
c	$\frac{dL_2}{d\theta}$	L_2(bits)	$\epsilon_{L_2}(\%)$	$\frac{dQ_2}{d\theta}$	Q_2(bits-seconds)	$\epsilon_{Q_2}(\%)$
0	-25279202.83	115719556.4	1.04034	-218471	8106657.755	0.67946
0.01	-24842045.23	115469394.3	0.83854	-216452	8104487.891	0.69043
0.02	-24416202.23	115223056.9	0.92157	-214458	8102338.315	0.40561
0.03	-24001772.32	114981145		-212494	8100202.438	
0.7	-15512557.04	104413352.6	-4.32811	-208894	7989693.713	0.17093
0.71	-16447356.35	104251513.1	0.05840	-208547	7987608.346	0.07320
0.72	-16554973.33	104087135.6	-0.35552	-208170	7985524.402	-6.69219
0.73	-16866033.31	103920997.3		-232441	7983303.393	
1.57	-24918729.33	83559239.85	0.12084	-200680	7735289.986	0.20482
1.58	-24861590.57	83310353.67	0.27094	-197704	7733287.292	0.88724
1.59	-24791224.33	83062411.37	0.21702	-197493	7731327.793	0.95061
1.6	-24672653.08	82815037.15		-195167	7729371.635	

$\frac{1}{T}\left(w_{L_1}L_1(\theta;T) + w_{Q_1}Q_1(\theta;T) + w_{L_2}L_2(\theta;T) + w_{Q_2}Q_2(\theta;T)\right) + kc$ where $w_{L_1} = 0.2, w_{Q_1} = 10, w_{L_2} = 1.5, w_{Q_2} = 2.5$ and $k = 320000$. A sample cost function (i.e., at seed = 1000) is shown in Figure 22a. It has a minima at $c = 0.27$. Using the sameseed for the 100 iterates of the optimization, we see that the algorithm does converge to $c = 0.27$ for all three initial values of c, i.e., $c = 0, 0.5, 1.4$. (See Figure 22b.) For the stochastic random case (Figure 22c), there is convergence at around a slightly higher value of c.

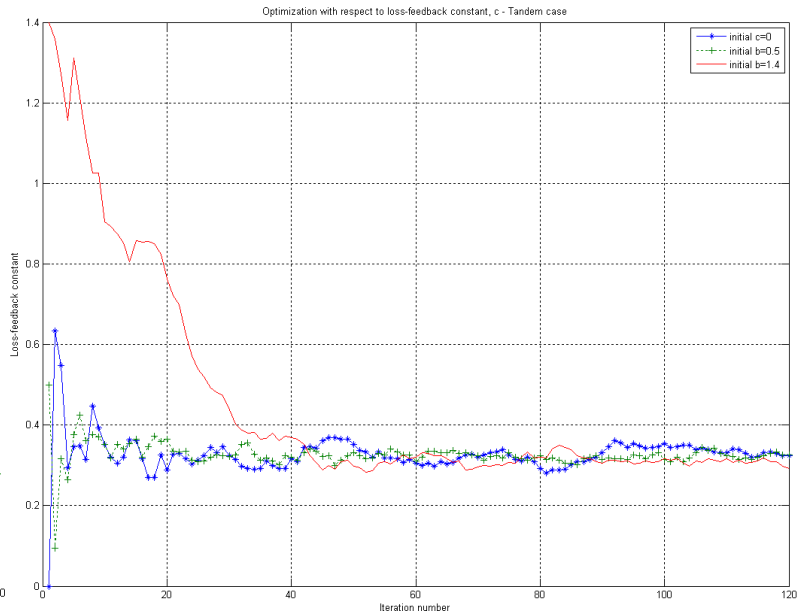


(a) Cost (seed = 1000)

Figure 22: IPA optimization with respect to the loss feedback constant, c - tandem case

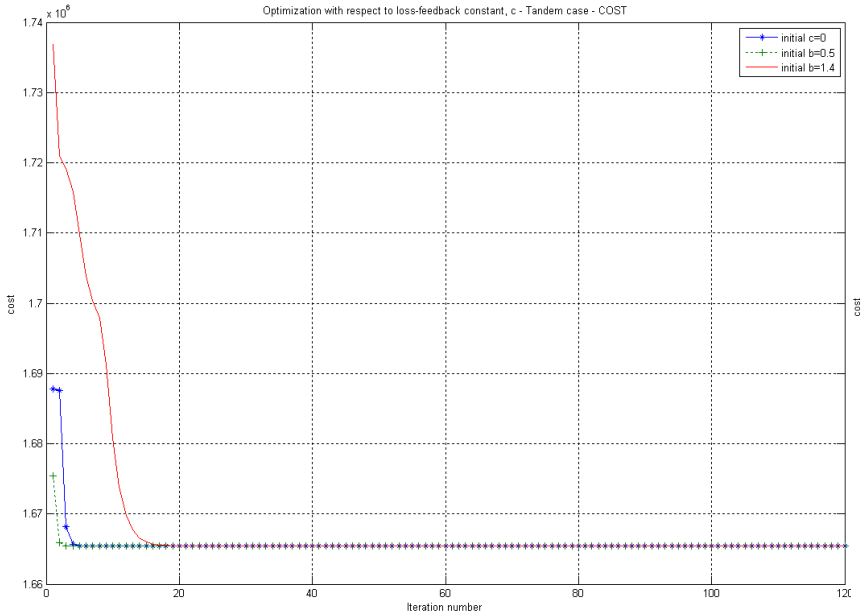


(b) Convergence - deterministic

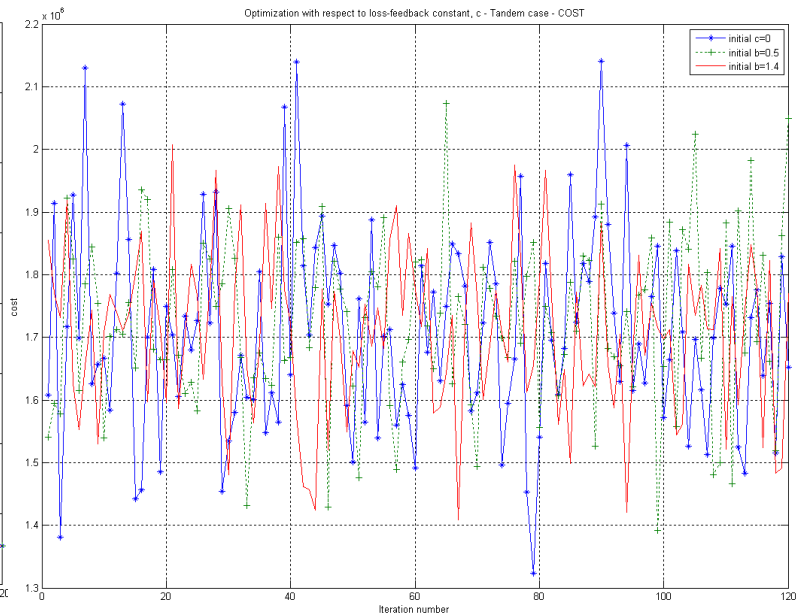


(c) Convergence - random

Figure 22: IPA optimization with respect to the loss feedback constant, c - tandem case



(d) Cost progression - deterministic



(e) Cost progression - random

Figure 22: IPA optimization with respect to the loss feedback constant, c - tandem case

CHAPTER VI

VALIDATION OF SFM/IPA IN THE DISCRETE MODEL

In Chapters 2 to 5, simulations were performed in a fluidized setting using Matlab so as to verify the accuracy of the IPA gradient estimators. Though derived in the SFM framework, we would like to see whether or not these same IPA-gradient estimators can be applied with reasonable accuracy in the discrete domain. The discrete model is more realistic for communication networks since the finest level of granularity is an entire packet. For queues in real communication networks, their lengths increase and decrease by one packet at a time. For the purpose of optimization, their capacity can only be incremented or decremented by at least one-packet at a time. This differs from the continuous flow paradigm in which the smallest change can be as small as a photon. Figure 23 compares the sample path of a queue in the discrete domain with that in the continuous flow domain. If an AQM scheme is to be developed using these SFM/IPA gradient estimators, it must be able to perform well in the discrete setting.

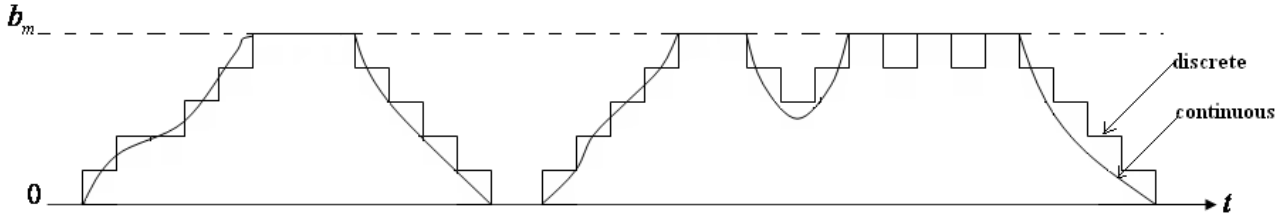


Figure 23: Trajectory of discrete model approximated by SFM

To this end, we perform simulations using the Georgia Tech Network Simulator (GT-NetS) [73]. It is a discrete-event packet simulator designed for communication networks. Each node in a GTNetS simulation (which can represent a host or router) is outfitted with queues, interfaces, and appropriate layers of the TCP/IP protocol stack, so as to emulate the behaviour of nodes in a real netowrk. The lowest level of granularity is the size of a

packet which, for the purpose of these simulations, is a constant value. We can therefore obtain a high-level of discretization in the GTNetS implementation.

The simulation parameters, test configuration and procedures outlined in Chapters 2, 4 and 5 are also used here in GTNetS. We consider only the case when buffer capacity is the control parameter. We present the results for the single-stage case and compare them to that obtained when Matlab was used.

6.1 Single-stage fluid queue with instantaneous, additive loss-feedback

Simulations were run for $T = 240$ seconds for three sets of four adjacent values of θ (in packets), namely $\{29, 30, 31, 32\}$, $\{50, 51, 52, 53\}$ and $\{99, 100, 101, 102\}$, for three different seed values for the random number generator, the error values are presented in Table 12.

For the group $b = [29, 30, 31, 32]$, the magnitude of the percentage error in loss derivative ranged from as low as 0.00024% to 2.8% (compared to that of Matlab which ranged from 0.00042% to 0.012%), and that of the queue workload derivative from 2.00% to 2.75% (compared to that of Matlab which was 0.15% to 0.16%).

For the group $b = [50, 51, 52, 53]$, the magnitude of the percentage error in loss derivative ranged from as low as 0.11% to 6.6% (compared to that of Matlab which ranged 0.005% to 0.15%), and that of the queue workload derivative from 0.77% to just over 3% (compared to that of Matlab which was 0.14% to 0.21%).

For the group $b = [99, 100, 101, 102]$, the magnitude of the percentage error in loss derivative ranged from as low as 0.275% to 5.65% (compared to that of Matlab which ranged from 1.10% to 2.72%), and that of the queue workload derivative from 0.92% to 3.81% (compared to that of Matlab which was 0.04% to 1.75%).

The optimizations were performed for different initial values of θ, (i.e. 20 packets, 70 packets and 200 packets). The number of iterations were 400. Their plots are shown in Figure 24. The same algorithm is used as in Section 2.3.2, however $a_0 = 20$ and $\rho = 0.6$. Again, the cost function that was to be minimized was $\frac{w_L}{T} L_T + \frac{w_Q}{T} Q_T$, but here $w_L = 1.0$ and $w_Q = \frac{4.0}{554}$.

For the deterministic optimization convergence occurred at around 35 packets which is the

Table 12: IPA sensitivity analysis - the single-stage case $\theta \equiv b$ (GTNetS)

	seed1					
b (pkts)	$\frac{dL}{d\theta}$	L(pkts)	ϵ_L(%)	$\frac{dQ}{d\theta}$	Q(byte-seconds)	ϵ_Q(%)
29	-592.502	87726	-0.25291	116.3439	1874165	2.57931
30	-581.252	87132	1.41961	116.4622	1940282	2.33872
31	-570.001	86559	-2.28045	116.4641	2006311	2.73382
32	-567.001	85976		116.7498	2072596	
50	-441.751	76946	-0.73546	116.9852	3261377	2.31786
51	-436.501	76501	-1.94705	117.1037	3327689	2.98503
52	-430.501	76056	6.62045	117.4479	3394501	0.77439
53	-423.001	75654		117.5623	3460071	
99	-276.001	59896	-4.34756	117.9654	6494160	2.63519
100	-274.501	59608	5.64687	118.1692	6561235	1.21325
101	-269.251	59349	0.46451	118.1388	6627495	0.92455
102	-267.001	59081		117.9425	6693549	
	seed2					
b (pkts)	$\frac{dL}{d\theta}$	L(pkts)	ϵ_L(%)	$\frac{dQ}{d\theta}$	Q(byte-seconds)	ϵ_Q(%)
29	-584.252	91959	1.92580	120.914	1943327	2.22231
30	-576.001	91386	0.00024	120.9354	2011802	2.09825
31	-570.751	90810	0.65727	121.12	2080206	2.25262
32	-566.251	90243		121.2086	2148818	
50	-442.501	81351	3.05109	122.6644	3390888	1.29337
51	-436.501	80922	-0.11430	122.5466	3459723	1.76045
52	-434.251	80485	-0.40274	122.5631	3528809	1.52142
53	-431.251	80049		122.6168	3597742	
99	-272.251	64227	-0.27522	125.3308	6804089	1.16333
100	-270.001	63954	4.81506	125.0798	6874330	1.32015
101	-267.001	63697	-3.37052	125.2721	6944539	3.26114
102	-266.251	63421		125.0755	7016203	
	seed3					
b (pkts)	$\frac{dL}{d\theta}$	L(pkts)	ϵ_L(%)	$\frac{dQ}{d\theta}$	Q(byte-seconds)	ϵ_Q(%)
29	-590.252	89778	1.90622	118.4976	1904748	1.81016
30	-585.002	89199	-0.68350	118.5827	1971584	2.63824
31	-579.001	88610	0.86380	118.5309	2039012	1.99019
32	-570.001	88036		118.7136	2105985	
50	-456.001	78831	3.72830	118.3653	3314930	1.68149
51	-444.751	78392	-4.10317	118.1618	3381607	3.00384
52	-440.251	77929	0.96561	118.2502	3449035	1.01875
53	-436.501	77493		118.1655	3515213	
99	-273.001	61650	-4.76164	120.2081	6600649	3.81065
100	-270.751	61364	0.64661	120.2148	6669782	1.90243
101	-268.501	61095	-4.28278	119.8297	6737648	2.38206
102	-267.001	60815		119.9152	6805615	

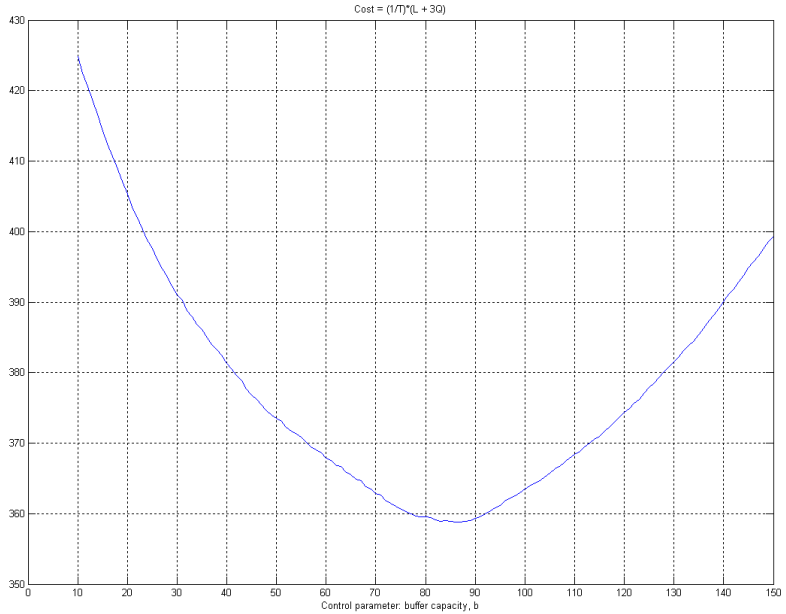
minima for the cost function evaluated using the same seed of the random number generator. For the randomized setting the optimization had variability, but the average was roughly around 30 packets which was slightly away from the minima.

6.2 Single-stage fluid queue with delayed, additive loss-feedback

Again, simulations were run for $T = 240$ seconds for three sets of four adjacent values of θ (in packets), namely $\{29, 30, 31, 32\}$, $\{50, 51, 52, 53\}$ and $\{99, 100, 101, 102\}$, for three different seed values for the random number generator, the error values are presented in Table 13.

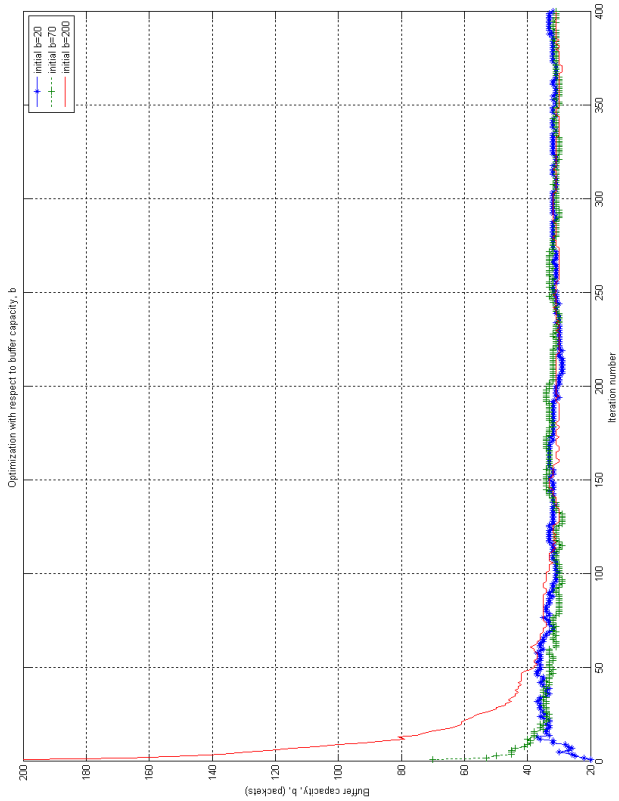
The optimizations were performed for different initial values of θ, (i.e. 20 packets, 70 packets and 200 packets). The number of iterations were 400. Their plots are shown in Figure 25. Using the same algorithm as in Section 2.3.2, with $a_0 = 1$ and $\rho = 0.6$, the cost function that was to be minimized was $\frac{w_L}{T} L_T + \frac{w_Q}{T} Q_T$ where $w_L = 10$ and $w_Q = \frac{40}{554}$. The time horizon for each run is $T = 10$seconds, and the loss-feedback delay is $T_d = 0.01$ second.

The cost function minima is roughly between 35 and 40 packets. For the deterministic case convergence occurred at 35 packets when initial buffer capacity was 20 and 39 (when initial buffer capacity was 70 packets). Convergence was much slower for the case when the initial buffer capacity was 200, nevertheless it approached very closely to 40. For the random case, convergence was slower with higher variability, however convergence did occur roughly between 35 and 40 packets.

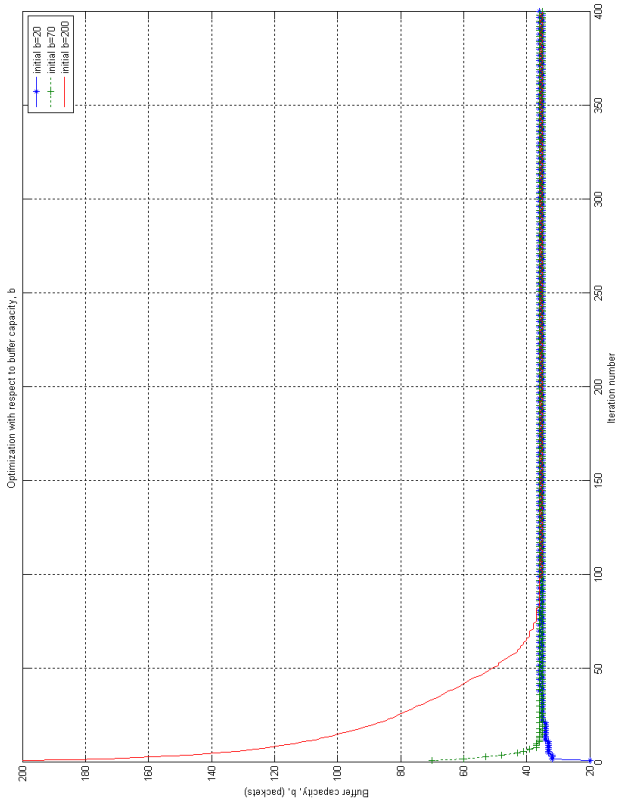


(a) Cost

Figure 24: IPA optimization with respect to buffer capacity, b - single-stage case

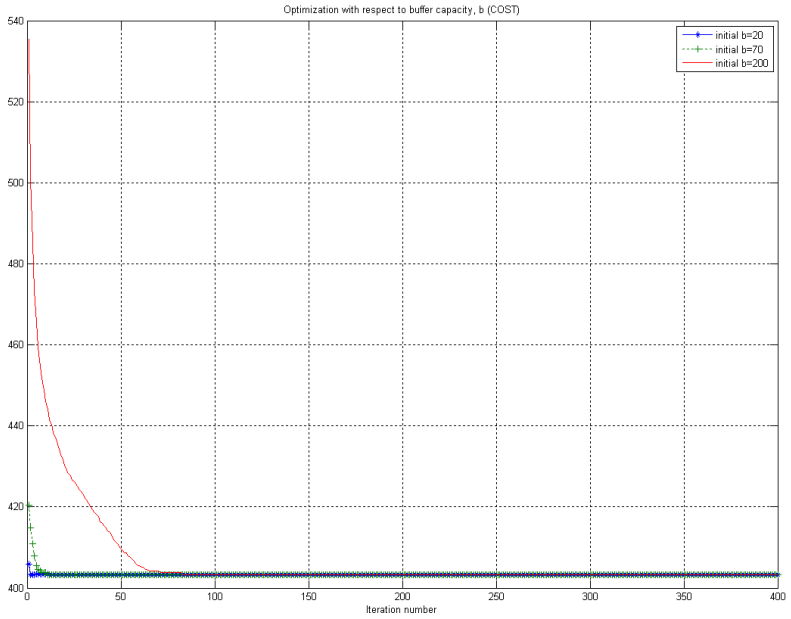


(c) Convergence - random

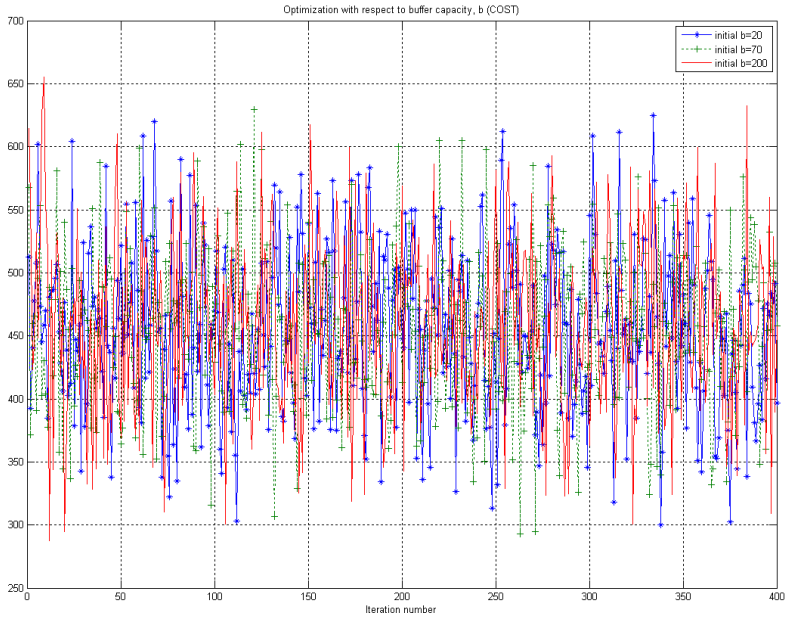


(b) Convergence - deterministic

Figure 24: IPA optimization with respect to buffer capacity, b - single-stage case



(d) Cost progression - deterministic

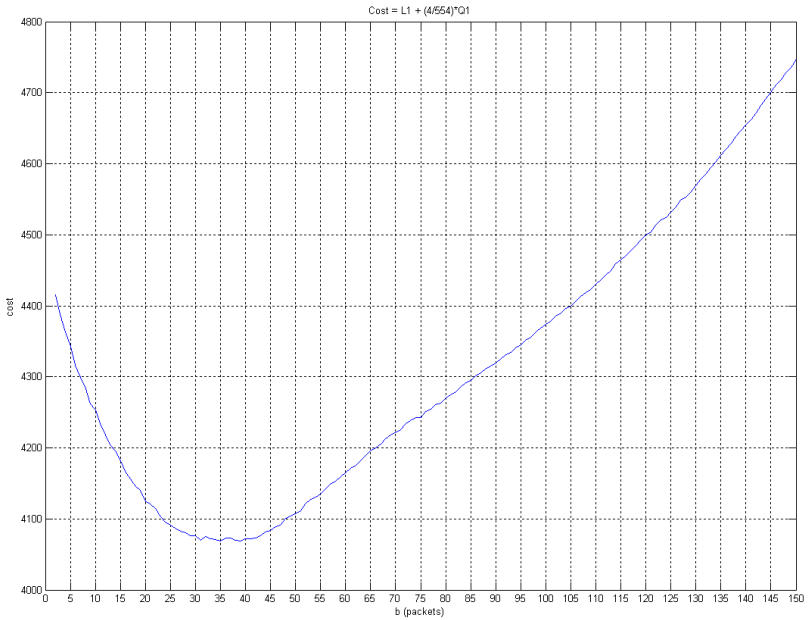


(e) Cost progression - random

Figure 24: IPA optimization with respect to buffer capacity, b - single-stage case

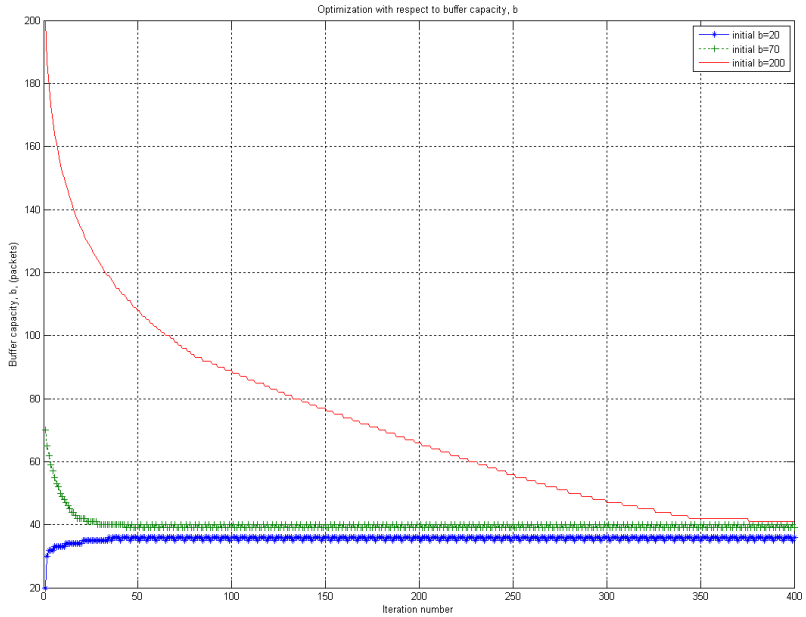
Table 13: IPA sensitivity analysis - the single-stage case $\theta \equiv b$ with delay $T_d = 0.01$s (GTNetS)

	seed1					
b (pkts)	$\frac{dL}{d\theta}$	L(pkts)	$\epsilon_L(\%)$	$\frac{dQ}{d\theta}$	Q(byte-seconds)	$\epsilon_Q(\%)$
29	-514.501	93672	3.40160	114.4328	1834669	2.59202
30	-505.501	93175	-4.45077	114.6808	1899708	3.51916
31	-498.751	92647	0.95262	114.763	1965477	1.79667
32	-492.751	92153		114.9115	2030198	
50	-402.751	84028	-2.04816	114.6411	3200087	2.42923
51	-397.501	83617	3.64804	114.6783	3265141	3.02718
52	-390.751	83234	-5.18207	115.0489	3330596	2.73924
53	-387.751	82823		115.1138	3396079	
99	-270.001	67719	-6.29602	113.3586	6404413	3.32534
100	-267.751	67432	4.76215	113.3555	6469302	1.04787
101	-266.251	67177	2.34767	113.1443	6532759	1.92728
102	-263.251	66917		113.2472	6596649	
	seed2					
b (pkts)	$\frac{dL}{d\theta}$	L(pkts)	$\epsilon_L(\%)$	$\frac{dQ}{d\theta}$	Q(byte-seconds)	$\epsilon_Q(\%)$
29	-510.751	97730	-3.18133	119.8498	1906689	2.73388
30	-500.251	97203	3.84833	119.8358	1974901	2.09969
31	-492.001	96722	-2.03227	120.2973	2042684	2.32471
32	-485.251	96220		120.3576	2110878	
50	-391.501	88325	0.38340	122.6657	3346178	2.54456
51	-390.001	87935	1.79512	122.6058	3415864	0.98992
52	-385.501	87552	-0.12944	122.7933	3484460	3.78452
53	-382.501	87166		122.9711	3555062	
99	-255.751	72794	0.29349	123.391	6774452	1.67702
100	-255.001	72539	3.13748	123.3524	6843957	0.72840
101	-253.501	72292	-6.11415	123.2071	6912792	5.90310
102	-253.501	72023		123.2328	6985078	
	seed3					
b (pkts)	$\frac{dL}{d\theta}$	L(pkts)	$\epsilon_L(\%)$	$\frac{dQ}{d\theta}$	Q(byte-seconds)	$\epsilon_Q(\%)$
29	-522.001	95841	1.53281	117.4396	1878878	2.38768
30	-514.501	95327	0.48616	117.1973	1945493	0.74344
31	-511.501	94815	-4.59406	117.1393	2010903	4.01082
32	-504.001	94280		117.31	2078401	
50	-390.001	86146	-1.02538	117.5934	3286021	1.23453
51	-386.251	85752	-0.71171	117.49	3351972	2.01805
52	-382.501	85363	-2.48339	117.4285	3418375	2.88464
53	-379.501	84971		117.4677	3485307	
99	-261.751	70328	1.05089	118.1438	6564595	1.34654
100	-261.001	70069	-1.53229	118.117	6630928	3.62056
101	-258.751	69804	-1.25580	118.0816	6698734	1.78668
102	-256.501	69542		117.7863	6765320	

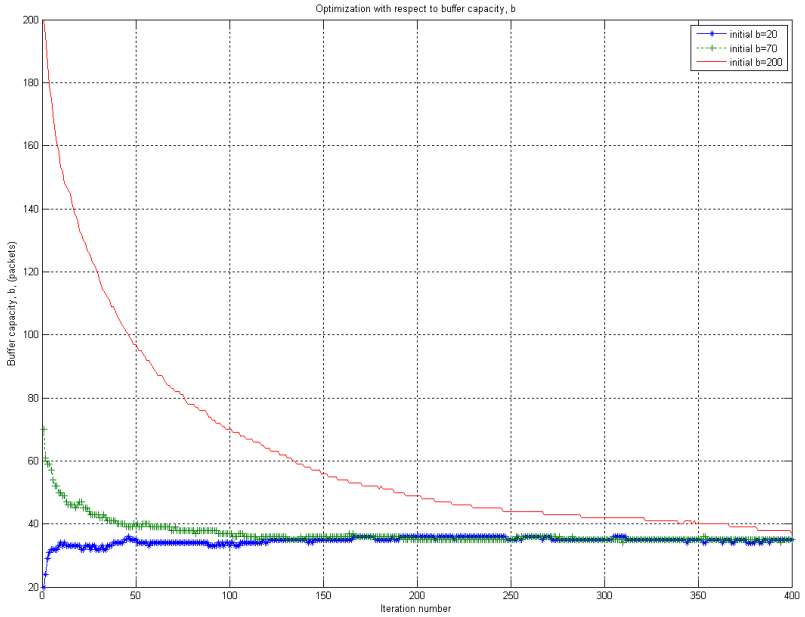


(a) Cost

Figure 25: IPA optimization with respect to buffer capacity, b - single-stage case with loss-feedback delay ($T_d = 0.01$ second)

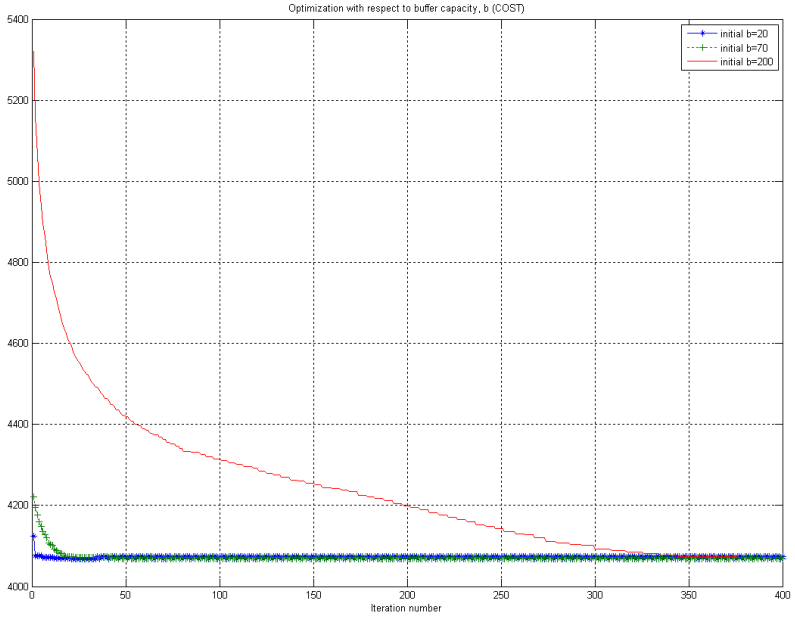


(b) Convergence - deterministic

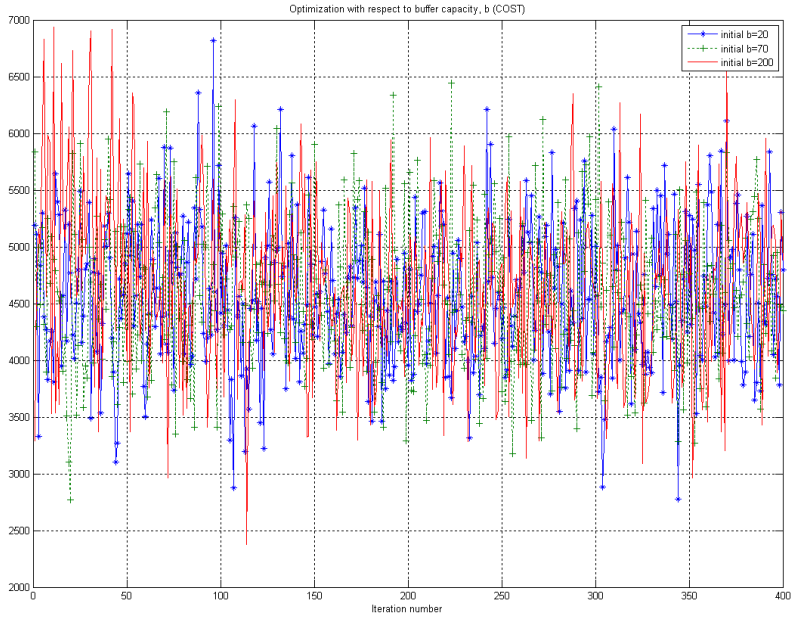


(c) Convergence - random

Figure 25: IPA optimization with respect to buffer capacity, b - single-stage case with loss-feedback delay ($T_d = 0.01$ second)



(d) Cost progression - deterministic



(e) Cost progression - random

Figure 25: IPA optimization with respect to buffer capacity, b - single-stage case with loss-feedback delay ($T_d = 0.01$ second)

CHAPTER VII

IMPLEMENTATION CONSIDERATIONS

In this chapter we discuss a number of issues that should be considered in the implementation of the IPA-based AQM in real-world routers. To begin, we briefly summarize the requirements of the IPA-optimization algorithm.

7.1 Requirements of the IPA algorithm

1. Each node must have a global (sufficient) view of the tandem of nodes associated with each flow passing through it. In particular, it must know the position of its own queue (corresponding to each flow) relative to the other nodes' queues in the tandem. Additionally, the tandem configuration for each flow must be fixed for the entire duration of the flow for there to be convergence to the optimum.

2. For each flow, every node in the associated tandem should know when certain events have occurred locally and at other queues and what these events are. These events include when the queue becomes empty, becomes full, ceases to be empty, and ceases to be full. They may at times also need to know the state of the other queues, i.e., whether they are empty or non-empty. Each node must have current values for the term $\xi_j(\theta; z_{m,i})$ for $j = 1, \ldots, M$, which was defined in an earlier chapter.

3. For each flow, every node in the tandem should be able to carry out the computation of the IPA derivative estimates for loss volume and queue workload.

4. There are times, when actual rate information must be used, such as inflow rate $\alpha_m(\theta; t)$ and $\beta_m(t)$. This is the case when the events are induced events. The node should be able to calculate these local instantaneous rates accurately enough and obtain accurate rate values from other queues.

5. Each node must know whether or not its buffer capacity is the control parameter. If

it is the control parameter, it must be able to carry out the adjustment in buffer-capacity (physically or virtually). To know how much the adjustment should be and in what direction (whether increase or decrease) will also involve a knowledge of the objective function for the optimization. It addition to this, it will require that the node holds a current value for the IPA derivatives computed at other queues.

6. For each flow, every node in the tandem should know what the various parameter values are, such as the loss-feedback constant, c, the performance weights, and the update interval (or time horizon), T.

7. Each source which has traffic passing through IPA-enabled queues should adjust their source rates according to the additive formula which is a function of the total loss-rate incurred along the tandem. This knowledge of the loss rate must somehow be physically fedback to the source to carry out the rate adjustment.

Before we embark upon strategies of how to fulfill these different requirements, we will discuss the basic structure, functionality and limitations of routers in the communication networks. Then, based on the requirements we have listed, we will itemize and discuss the main tasks (with their related challenges) that should be carried out at the nodes, at the source and at the destination. We then offer suggestions as to how to reduce the complexity of the algorithm. We finally consider how this IPA-based AQM implementation can be incorporated into the proposed Internet Quality-of-Service (QoS) frameworks such as IntServ and DiffServ.

7.2 Routers

The authors of [43] summarized the major duties of routers in communication networks as follows:

- To transfer packets at their input ports to the appropriate output ports based on routing tables and the destination address of packet.

- To participate in distributed routing algorithms so as to generate accurate and up-to-date routing tables.

- To translate, when necessary, between different link technologies, e.g., Ethernet, PPP, and so connect networks that use different protocols.

- To perform scheduling so as to provide some measure of priority or QoS guarantees to flows.

Besides these four responsibilities, routers are called upon to participate more actively in congestion control using Active Queue Management (AQM), and it is here we seek to position the IPA algorithm. But we must first determine the modules in the router that will actually participate in the AQM function. Also, we need to determine those factors that already limit the speed of routers and impact their cost, and how the IPA algorithm would, in turn, impinge on that speed and cost. This is especially important for backbone routers which interconnect hundreds of smaller networks and must therefore operate at very high-speeds and for enterprise routers which must be low-cost.

A generic router consists of four main components: input ports, output ports, the switching fabric and a routing processor [43]. Here, we assume that the router is output-queued, i.e., the switching fabric has a bandwidth that is greater than the total bandwidth of the input ports so that there is no queueing at the input ports but rather at the output ports.

The input port is the entry point to the router for packets from an incoming link. It is located on interface/line-cards. Besides data link layer encapsulation and decapsulation, and possibly packet classification for QoS guarantees, its chief function is route lookup and forwarding. Based on the destination address of a packet, it determines from the forwarding table stored in its memory the output port to which the packet should be forwarded. It then sends the packet through the switching fabric. According to [43], the main bottleneck in backbone IP routers is the time taken to do route lookup by the input port as it cycles through the many entries of the large forwarding table. This time is a function of the number or memory accesses to find a matching route and the memory access time. As a result, the more packets that a router must forward, the more route lookups an input port must perform, which implies greater processing delays.

The switching fabric (also known as the interconnection network) provides the connection between the input and the output port. There are three types of switching fabric: the shared-memory, the bus and the crossbar. For the bus implementation, the bus contention limits the speed of the switching fabric. For the crossbar switching fabric, a scheduler would be required to turn on or off crosspoints. Hence it is the speed of the scheduler that limits the speed of the switching fabric. For the shared-memory (for which the packet is copied by the system's CPU), the speed of memory access is the limitation.

Upon reaching the output port, the packet is queued before being transmitted over the outgoing link. It is here that the packet will be subject to scheduling algorithms and AQM. It is here also that we have the issue of queuing delay and packet loss due to buffer overflow. The output port also performs data link layer encapsulation and decapsulation. One bottleneck at the output port is the DRAM/SRAM access time for the output queues so as to read out the packets for transmission. This limits the speed of the output-queued router [43].

Scheduling at the output port provides a means for applying priorities among flows or groups of flows. It contributes to the realization of QoS guarantees in networks. For the scheduling mechanism, each flow (or aggregate of flows) is assigned its own queue and is thus isolated from the others. The scheduler then serves each flow (queue) according to some priority order and/or weighted scheme. Examples of such scheduling mechanisms: Weighted Fair Queuing (WFQ), Dynamic Round Robin (DRR). In real routers, there is a limit to the number of individual flow queues per port for scheduling. It will be impossible to assign a queue to each individual flow, the number of which can be in the order of millions [45], and this will require astronomical amounts of memory. As a result, aggregate of flows (possibly based on traffic type - TCP, Multimedia, etc.) are queued separately instead.

The routing processor runs routing protocols and participate in route calculations. It computes the routing tables which are then used by the input ports. It runs software to configure and manage the router. It also handles anomolous packets (e.g., destination address cannot be found). The time taken to perform routing updates can also affect router speed. However, the authors of [43] (who cited [51]) mentioned that routing tables needed to be

updated only once every two minutes.

So far we see that route lookups, switching fabric technology and memory access to output buffers are the main bottlenecks to router speed. We now look at one of the main contributors to router cost, i.e., the cost of a port. The cost of a port depends on its processing power, the amount and kind of memory it uses and the complexity of communication between the routing processor and the port [43]. Processing in ports can be performed by general-purpose processors or ASICs (which are cheaper and faster). However, general-purpose processors offer more functionality. In terms of types of memory, DRAM tend to be cheaper but slower than SRAM. Therefore SRAM is used for backbone routers, whereas, DRAM for enterprise and access routers [43]. Also larger buffer sizes imply greater up-front memory cost. In terms of communication complexity, the tradeoffs between centralized processing or distributed processing should be considered. It may be necessary to determine which functions be delegated to the routing processor and which to the port. If more functionality is to be alloted to the port, the likelihood of needing a general purpose processor increases. However, if the router processor must carry out most of the work, then the communication between itself and the ports intensifies as it issues commands to the ports and receive data from the ports. This will require more sophisticated internal communication protocols.

7.3 Output queue structure for IPA

The IPA algorithm can be a purely software process installed in the output port and will require its share of processing memory. The derivation of the IPA algorithm for the tandem case is on a per-flow basis. So, it may be assumed immediately that some form of scheduling will be required among the IPA flows, i.e., each IPA flow will be assigned its own queue. However, the reason for scheduling in general is to ensure priorities among flows. We contend that this is not the main aim for the isolation among the flows that are IPA controlled. The main purpose for the isolation is so that the IPA control can be carried out more efficiently for each individual flow since each flow may have its own set of parameters. In fact, to carry out the IPA algorithm the trajectory of queue occupancy for each flow is required.

We therefore suggest the following structure at the output queue:

- Instead of each IPA flow having its own queue, have only one queue assigned to all IPA flows, and another assigned to non-IPA flows. This will require some form of classification to distinguish non-IPA from IPA flows at the input-port. The Type Of Service (TOS) field in the IP header can be used for that purpose. The scheduling between the two queues could use any priority scheme. However packets within the queues themselves are serviced according to the first-come-first-serve (FCFS) procedure.

- In order to distinguish the different queue occupancies for the different IPA flows, we assign to each flow a variable called the virtual-queue length (i.e., $VQlen_i$ for flow i), which holds a running count of the number of packets belonging to that flow that has been queued in the general IPA output queue. For each flow, the virtual-queue length has a minimum value of zero, and a maximum value (which is what was called the buffer-limit, b_m, in the derivations of the IPA derivatives in previous chapters). The virtual-queue length for a flow is incremented every time one of its packets arrives and is enqueued at the general IPA queue, and it is decremented every time one of its packets departs the general IPA queue. However, when the maximum value of the virtual-queue length is reached, the flow has reached its buffer-limit (in terms of queue occupancy) at the router. Any packet (of that flow) that arrives after this point is considered "lost". The packet can be dropped (erased) from the general IPA output-queue or it can be marked by setting the Explicit Notification (ECN) bit in the IP header.

Figure 26 illustrates the IPA output queue structure and shows the current virtual queue length for three IPA flows currently in the general IPA output queue.

There may be times when rate-information is needed for the IPA algorithm, in particular, the instantaneous arrival rate for the flow at the router and the instantaneous service rate for the flow at the router when the flow's virtual-queue length is non-zero (i.e. queue non-empty). The arrival rate can be measured as the inverse of the time-interval between consecutive packet arrivals of the flow at the router. The derivation of the IPA algorithm

assumed general, time-varying service rates. The service rate would be the inverse of the time-interval between consecutive packet departures for that flow from the general IPA output-queue.

A complication arises when the virtual-queue length for the flow is less than its maximum value, but its packets arrive at the general IPA output queue when the latter is full. In this case, they will be literally dropped. One possible approach would be to freeze the virtual-queue length which would imply that the packets never arrived. However, the source would have detected that losses did occur and reduce its rate in response. Another possible approach would be to set the virtual-queue length to the maximum, inferring that a loss event has occurred, and so the IPA algorithm will treat it as such.

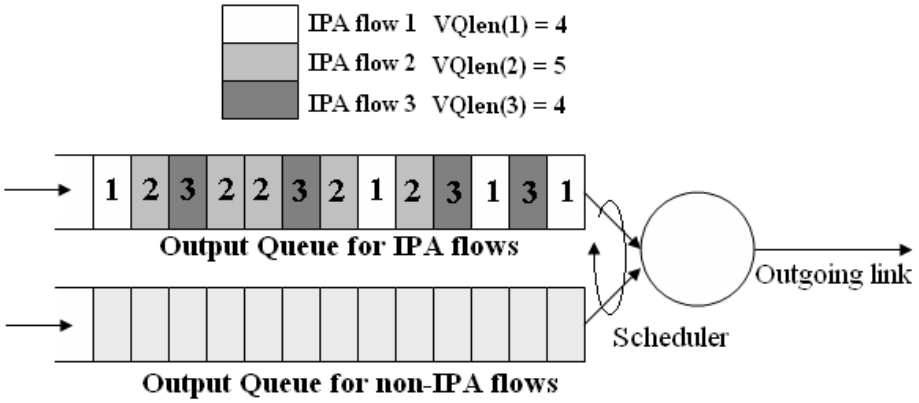


Figure 26: Example of output queue structure for IPA

7.4 Main IPA tasks at routers

To deal with the first requirement, the route must first be established, afterwhich each router in the tandem must have a stored copy of that route. For the second and fourth requirement, there must be some way for the other routers to inform one another about events occuring locally at their output queues and about their local inflow and service rates. We will call this task "Exchanging Information". Based on all the information collected from other queues and locally, a router must then calculate the IPA derivatives and adjust the control parameter when necessary. These tasks cover the third and fifth requirements.

We now elaborate on these tasks.

7.4.1 Establishing the route

Currently, there are two main classes of routing protocols in the Internet: distance-vector routing and link-state routing. By distance-vector routing, routers exchange their routing-tables with neighbouring routers, and using this information together with the Bellman-Ford algorithm, each router calculates new shortest-paths and update their routing-tables. In this class of routing and for a given flow, a router does not know what the other routers are along the source-to-destination path to which it belongs. Instead, each router just has tabulated every other node in the network with the corresponding next-hop (or neighbour) node to get there. Examples of routing protocols based on distance-vector routing, include Routing Information Protocol (RIP) and Interior Gateway Routing Protocol (IGRP).

Link-state routing is computationally more complex than distance-vector routing, but require less overhead when exchanging routing information. Using a reachability protocol, routers determine who are their immediate neighbours. They then flood the network with link-state advertisements telling all the other routers just who their neighbours are. In this way, each router can independently build a graph of the entire network. Based on this topology and using Dijkstra's algorithm, a router computes the shortest path from itself to any other node in the network. From the shortest path calculation, the router then builds its routing table which consists of every destination router and the next-hop router to get there. We see that for link-state routing a router has a total view of the network, a requirement for the IPA algorithm. But it only has the shortest path calculation from itself to other nodes. For IPA, the router must also know its position (when necessary) in the sequence of routers comprising the shortest-path between every source-to-destination pair of routers to which it belongs and for which it is not the source router (i.e., $m \neq 1$). It may perform this calculation independently based on the topology it has for the network, or it may receive shortest-paths already calculated by other nodes through message exchange.

Open Shortest Path First (OSPF) and Intermediate System to Intermediate System (IS-IS) are two routing protocols based on the link-state routing approach. OSPF, which is

the most widely used interior gateway protocol (IGP) is hierarchical. A network that uses OSPF is usually divided into areas. One of these areas form the backbone/core via which and only through which all other areas can commumunicate with one another. A router within a given area can build a complete topology for its area only. It cannot distinguish individual routers belonging to other areas. As a result, a router within a given area will not know the complete end-to-end path that transcends areas. This is a challenge for the IPA algorithm.

Therefore, it may be necessary to augment current routing protocols to support full topology information at each node. For example, each area could have a designated router that sends messages to other designated routers belonging to other areas with information about the topology within its own area. These routers, in turn, share this information with those routers within their own area.

In both cases (i.e., distance-vector and link-state routing), shortest paths must be re-computed and routing tables must be updated when the network topology changes (e.g., due to link failures). If these topology updates occur during the time the IPA derivatives are being calculated, then the IPA algorithm should be aborted. After the new topology has been established, then should the IPA algorithm be re-started.

We have not considered exterior gateway protocols such as BGP, since they manage routing outside of a given autonomous system. For now, we limit the scope of IPA to within an autonomous system.

7.4.2 Storing the route at each router

In addition to the traditional routing tables (for the purpose of forwarding packets), each router should store a copy of the entire network topology even after that router has performed its own shortest path calculations and has carried out its own routing table updates. This is for when the router has to determine its position in the route for on-demand source-destination paths in order to execute the IPA algorithm. For each new source-destination pair of routers (identified by the IP addresses of both the source router ($m = 1$), and the destination ($m = M$)), the router should have stored its position number, e.g., $m = 3$, so

that in performing the IPA algorithm it will be able to distinguish messages coming from upstream nodes (e.g., $m < 3$) from those coming from downstream nodes (e.g., $m > 3$). We use the term "on-demand" to suggest that the position number need not be stored for all possible source-destination paths. It is only when a flow requiring a new source-destination path appears, then should router determine its position number. Additionally, one can impose a time limit on how long a source-destination-path/position-number be stored beyond the life of a flow that had need of it.

7.4.3 Exchanging queue information

The computation of the IPA derivatives at any given node (i.e., router) requires the knowledge of certain events occuring at the queues of other routers. To facilitate this, messages containing information about these queue events can be exchanged among nodes in a timely manner. In particular, each of these messages should contain the source-destination-routers identifier, the flow identifier, the position of the node (at which the event occurred), the type of queue event (i.e. become-empty, cease-to-be-empty, become-full, cease-to-be-full), the time of occurrence of the event and the derivative term $\xi_j(\theta; z_{m,i})$. Depending on the event-type, the message may also contain the net-inflow rate at the queue at the time of the event. Additionally, depending on the event-type, a node must decide to which other nodes it should issue the message. The easier but more resource-consuming approach would be to have the node broadcast the message to every other node on the tandem. Each node then decides whether to use or ignore the message. A more complex method, would be to have the originating node decide exactly which set of nodes should receive the message. For example, a queue-becomes-full event only affects upstream nodes. So when such an event occurs at a given node, it should immediately send that message to the first non-empty node in the tandem. But how does it determine a priori which is the first "non-empty" node? This will require that the node have state information (i.e., whether the queue is empty or non-empty) for queues at nodes that are upstream from itself. Therefore, a compromise position could be that the node send messages of that event-type to all upstream nodes only. The upstream nodes then determine among themselves which is the first non-empty

node using a light-weight queue-state message exchange. The elected node then processes the message, while the others ignore it.

One possible design question is whether these messages should be sent over a separate signaling path or as "dummy packets" along the same path as data packets. The former is an expensive option since it involves reserving (on a permanent basis) resources for the sole purpose of network management. The latter option may be comparatively costly depending on the number and size of the messages and the frequency of the message exchange. These factors also impact the processing requirements at the nodes. This leads to the other design question: "How often and when should these message exchanges occur?" Ideally, as soon as a queue-event occur at a node, the message should be sent. But if nodes generate queue-events with high frequency due to the current dynamics of the network, the message load can become prohibitively high. A less accurate alternative would be to periodically send messages containing batches of queue-events and to also impose a maximum number of occurrences of such queue-events that will be shared with other nodes. A compromise position would be to have nodes send queue-events as soon as they are generated when the frequency is low, but above a certain threshold resort to the periodic batch method. This frequency regulation can be implemented by a token bucket of sorts. Simulation would have to be carried out to see the effect of this method on the accuracy of the IPA derivatives.

Two other issues with exchanging queue information by messages are the effect of lost messages and the effect of delayed messages. These two can also negatively impact the accuracy of the IPA derivatives that are calculated at a given node.

7.4.4 Calculating the IPA derivatives

The IPA derivatives are calculated on a per-flow basis, where the sample path of the virtual-queue length for each flow is examined separately. The actual computation of the IPA derivatives can be carried out as separate independent software processes for each virtual-queue length, where each virtual-queue length is associated with a different flow/tandem. (See Section 7.3.) These software processes themselves require processing and memory resources. The IPA computation consists of two main but interlocking parts, the recursive

algorithm and the computation of the IPA derivatives. From now on, we will refer to "virtual-queue length" simply as "virtual queue".

To participate in the recursive algorithm among the virtual queues associated with a flow, the node must keep track of the boundary times and queue-events of the flow's virtual queue, and in turn send messages to other nodes about these events. Upon receiving queue-event and queue-state messages from other nodes, the node stores the information from these messages according to their flow identifier so as to execute the IPA algorithm for each flow. For each flow identifier, and for each queue-event associated with the flow, the state of each virtual queue belonging to the tandem (and associated with the flow) is stored, including that of the node itself. In other words, M bits of memory can be allocated for each flow (where M changes from flow to flow), where the j-th bit indicates whether or not virtual queue $j = 1, \ldots, M$ is empty or not empty. For each flow identifier and for each queue-event associated with the flow, the node must store the variable $\xi_j(\theta; z_{m,i})$ for $j = 1, \ldots, M$ where $z_{m,i}$ is the event time. These variables are then used in both the recursive algorithm and the computation of the actual IPA derivatives at the node. The memory associated with the different variables of $\xi_j(\theta; z_{m,i})$ can be overwritten after the occurrence of each queue-event at the flow's virtual queue at the node in question/view.

There is the occasion when actual net-flow-rate information may be needed. This is mainly due to induced events. Flow-rate information will be required from the inducing queue-event (in which case the rate information would be in the queue-event message), and from the node at which the queue-event was induced (in which case the node must be able to measure directly the net-flow-rate). Provided that these induced events occur with low probability, they can be ignored, and the rate computation can be omitted with very little change in IPA accuracy.

7.4.5 Adjusting the control parameter, b_m

Since the virtual queue for a single flow is really a running count of the number of packets associated with that flow that are currently in the physical queue at the node in question, its limit (b_m) is stored as an integer variable. If that limit is reached the virtual queue is

considered to be full. The actual physical packet that arrives when it is full is dropped from or marked (using ECN) in the physical queue. This limit, if it is the control parameter, can be easily adjusted by a change in the value of the integer variable. However, a minimum and maximum value for this limit should first be specified so that the new limit after adjustment remains within these bounds.

Additionally, the node (and more specifically, the IPA algorithm associated with the virtual queue of the flow at the node in question) must know the objective function for the flow in order to perform the adjustment in the correct direction (e.g., an increase or a decrease in b_m). The objective function may involve sample performance functions from the other queues along the flow's tandem. Consequently, the node may also need the corresponding derivative estimates from these other nodes.

The initial value of the limit before optimization is a design choice. So far there are no design rules for this value. For faster convergence, it should be close to that optimal value of b_m. But to estimate, a priori, where that optimal value just may be, can be a subject of future research.

7.5 Main IPA tasks at source and destination

In response to losses, the source must reduce its rate, $\sigma(t)$, according to the congestion control equation (e.g. $\sigma - c\sum\gamma$). But, it must first accurately detect these losses. We discuss in more detail these two main tasks for sources, i.e., detecting losses and adjusting source rates.

7.5.1 Detecting losses

Sources using the Transmission Control Protocol (TCP) indirectly detect losses via acknowledgments packets from the destination. Upon repeatedly receiving an acknowledgment packet with the same sequence number (plus one) of the last in-order packet correctly received by the destination, it determines that the packet is lost. Because TCP is a reliable protocol it resends that packet specified by the sequence number. For the case of IPA sources, it may not be necessary to know exactly which packets were lost, but rather how many were lost and the time-interval between losses, so as to calculate the loss rate. This

is true for TFRC. Sequence numbers will still be required so that the destination can know that there are missing packets within a flow, and just like TFRC, out-of-sequence packets are also treated as lost packets. Also, just like TFRC, acknowledgment packets will be required to inform the source of the losses.

The design issue, however, is the frequency at which these acknowledgment packets should be sent to the source. Ideally, as soon as a loss is detected by the destination, it should send an acknowledgment packet indicating such back to the source. Another approach would be to have the destination compute a running average of the loss-rate and periodically send this loss-rate information to the source using acknowledgment packets. Note, however, that the algorithms for the IPA derivatives were based on instantaneous loss-rates and not on average loss-rates. Hence, the accuracy would be affected. Future research could be to see by how much it is affected.

As a case-study for the latter approach, we can consider TFRC. In TFRC, a single loss-event consists of one or more detected losses within a round, where a round is the time between acknowledgments received at the sender. These losses can be packets that were not received, packets with the ECN bit set, and packets that are out-of-sequence.

Upon receiving a packet, the TFRC destination checks to see if there are any missing packets between the one that was just received and the one that was previously received. It does this by examining the sequence number of the packet. If the packet received has its ECN bit set, and it is deemed the beginning of a new loss-event, the "ecnEvent" flag is set. If the packet does not have its ECN bit set or it is not deemed the beginning of a new loss-event, TFRC attempts to determine the status of those missing in-between packets. A missing packet is considered to be the start of a new loss-event, if the difference between the time it was estimated to arrive and the beginning of the last loss event is greater than the round-trip time (rtt), and the round is a new round. When these conditions hold, the "congestionEvent" flag is set.

When either the "ecnEvent" flag or the "congestionEvent" flag is set, the destination immediately packages an acknowledgment to the TFRC sender. This acknowledgment packet has, among other things, an estimate of the loss-event rate. The acknowledgment packet

also has a count of the number of losses since the last report (or acknowledgment) to the source.

To estimate the loss, the destination uses the Weighted Average Loss Interval (WALI) algorithm. The number of packets between the beginning of each loss-event are counted. Each count is stored as a separate sample, and the weighted average across the most recent n samples is computed. This is taken as the average interval (in terms of packets) between loss-events. The loss-event rate is then reported as the inverse of this average interval.

It should be noted that if the destination does not receive a data packet after a certain time (e.g., $1.5 \times rtt$) then it generates an unsolicited acknowledgment to the sender, which will also hold an estimate of the loss-event rate.

The source, upon receiving the acknowledgment, extracts the estimated loss-event rate, and using the TCP throughput formula, calculates its new theoretical sending rate.

7.5.2 Adjusting source rates

How the source performs the adjustment $\sigma - c\sum\gamma$, depends on how the loss-rate was fed back to it from the destination. If explicit loss-rate values were computed then the source simply applies the formula. However, if acknowledgments for loss packets are instead sent, then the source can keep a running count of number of consecutive losses of which it is notified. For every $\lfloor \frac{1}{c} \rfloor$ lost packets, the source drops one (1) packet from its stream or it increases the interval to the next packet-transmission time by one packet-time.

For the latter approach, the adjustment of c as the control parameter becomes problematic. Uniform increments in c does not translate into uniform increments of $\lfloor \frac{1}{c} \rfloor$. Additionally, it is preferred that c be such that $\frac{1}{c}$ is a whole number. For example, if $c = 0.3333$, then the source drops one packet from its stream for every three lost packets.

7.6 Parameter choices

Each flow (and hence each tandem of nodes) has its own set of parameters: c, T, wL_j and wQ_j for $j = 1, \ldots, M$ (where M is the number of nodes in the tandem for the flow). Consequently, the source must share with the tandem nodes the values of these parameters for proper execution of the IPA algorithm.

The choice of each parameter affects the behaviour of the IPA-controlled flow, and depending on the difference in the values of these parameters for these flows, can affect the fairness among flows.

7.6.1 The loss-feedback constant, c

The loss-feedback constant, c, determines how aggressively a flow will reduce its rate. The higher its value, the more drastically it will reduce its source rate for a given loss-rate. However, unless c is the control parameter, it can be arbitrarily and independently set by the source. It is then possible, that IPA sources with different values of c can unfairly compete for resources at nodes.

7.6.2 The time-horizon, T

The IPA derivatives are computed over the time T, and for the optimization, the control parameter is adjusted every T. Therefore, if it takes n iterations for convergence to the optimal value, the time required to reach there would be nT. As a result, T directly impacts the overall speed of convergence. If T is too large compared to the variability in network conditions (e.g., the frequency of failing nodes), then convergence will never occur. On the other hand, if T is too small, the computed IPA derivatives, which are really estimators, will be too noisy. A trade-off is therefore required. One rough measure of the variability in network conditions would be how often routing protocols in networks must do routing updates. Choosing a value of T such that nT is smaller than the time between updates, may be a good heuristic.

It should be noted that n itself depends on the initial value of the control parameter used, the form of the objective function and the nature of the gradient estimator itself. If the initial value is far from the optimal, it will take a longer time for the algorithm to converge to the optimal value.

7.6.3 The weights wL_j and wQ_j for $j = 1, \ldots, M$

So far, the performance weights have been used when the form of the objective function is of the form $\sum_j wL_j \frac{dL_j(\theta;T)}{d\theta} + \sum_j wQ_j \frac{dQ_j(\theta;T)}{d\theta}$. These weights can be arbitrarily set by the

source and communicated to the flow's tandem of nodes. There is no direct translation of these weights to actual values of QoS parameters (such as loss and delay). Therefore, using these weights, one cannot pre-specify QoS parameters. As mentioned in the Section 7.3, the primary focus of the IPA algorithm is to provide efficient congestion control. It does not seek to provide to an IPA-controlled flow absolute QoS guarantees (i.e., a specific value of loss, delay or jitter). By having the algorithm optimize according to the objective function, the flow experiences the "best" possible performance (in terms of loss and delay) given the current situation (of network load and topology). By including weights in the objective function, the source can still have some direct influence on the relative importance it places on loss and delay at various points along the flow's tandem.

For QoS, the sources first agree to the nature and amount of traffic that they will inject into the network, afterwhich they must be kept conformant to that promise. For congestion control, however, the sources do not make an agreement upfront, but is restrained so as to keep the network highly utilized and stable. As a result the performance experienced by sources under congestion control can be variable. For congestion control, the concern is for the stability of the network as a whole and not primarily for the welfare of individual flows. If actual values of QoS parameters were required at the outset, then some form of resource reservation, call admission control, and traffic policing must be implemented. These surpass the congestion control functionality of the IPA algorithm. However, it should be noted that the scheduler, by separating IPA flows from non-IPA flows so as to carry out the IPA algorithm, provides some measure of isolation or protection to IPA flows. Priorities can be set between IPA flows and non-IPA flows in general.

7.7 Reducing complexity

We now suggest some ways to reduce the complexity of the IPA algorithm. Two main contributors to the complexity are the queue-information exchange and the amount of state information held for each flow at each node.

To limit the queue-information exchange, only a subset of queues can be allowed to communicate at any given time. For example, when downstream queues become full, an upstream

queue, Q_m, only feels the effect of this event when all queues before it (i.e., $j = 1, \ldots, m-1$) are empty. The likelihood of all queues, $Q_j, j = 1, ..., m-1$, being empty at the same time decreases significantly as m increases. Therefore, when such an even occurs, one can limit the exchange only to the first few queues (say, two) in the tandem (i.e., $j = 1, 2$). Similarly, when an upstream queue, Q_i becomes empty, a downstream queue, Q_m, is affected only when the queues that are in-between, i.e., $Q_j, j = i+1, \ldots, m-1$ are empty. Again, the likelihood of all these queues being simultaneously empty decreases as $m - i$ increases. As a result, information exchange from Q_i can be limited to say, Q_{i+1}, Q_{i+2}. The consequence of limiting queue-information exchange would be a drop in accuracy. However, further studies will be needed to determine if the drop in accuracy is justified by the reduction in complexity.

Instead of having each flow be assigned its own virtual-queue length, parameters (e.g., T, c and weights) and variables, we can have aggregates of flows be assigned common virtual-queue length, parameters and variables. The depth into the network at which this aggregation can begin (i.e., from the perspective of the IPA algorithm) is a design choice. For example, if IPA is performed only in the backbone portion of a network, then those routers on the periphery of the backbone must peform function $\sigma - c\sum\gamma$. The sources, themselves, do not get the IPA-loss feedback, but rather those routers on the periphery of the IPA zone. However, communicating losses within the network itself (i.e. among routers), will be more challenging than for the traditional end-to-end archetype.

To a lesser extent, the complexity of the IPA algorithm also hinges on the complexity of the objective function. One way to lower this complexity is to limit the number nodes along the tandem that actually perform buffer adjustments, to, for example, the first node only.

7.8 The Internet QoS framework

As the heterogeneity of applications increases in the Internet so is the demand for QoS [50, 41]. FTP and email cannot tolerate errors, but do not have delay constraints, whereas VoIP, video and other real-time and streaming applications have very tight delay and jitter requirements, and are more error resilient [16]. If the Internet must simultaneously support

these different applications in an acceptable way, it must be re-engineered to meet their different QoS demands. Right now, the Internet does not make any QoS committments to users but serve them as best it can given the current conditions, i.e. 'best-effort service'. Its main focus is congestion control. The IPA algorithm that was outlined is a more advanced form of congestion control geared more to multimedia flows. However, of itself, it does not provide absolute QoS guarantees.

By taking the issue of fairness and flow-differentiation to another level, some researchers have been incorporating AQM into architectures that will provide actual QoS guarantees to Internet users. There are two main proposals for Internet QoS: IntServ and DiffServ. The DiffServ proposal has AQM as a vital player in QoS provisioning/enforcement.

7.8.1 IntServ

For each flow, resources must first be reserved at all the routers along the path, before its traffic is allowed into the network. To carry out this per-flow reservation, the signaling protocol, Resource Reservation Protocol (RSVP) is used. (See RFC 2205.) IntServ provides per-flow QoS guarantees. According to [71], IntServ does not scale well with increasing flow-count since the RSVP signaling is complex, and state-information must be maintained at the routers for each flow while it exists.

7.8.2 DiffServ

At the network edge, flows are tagged according to Service Level Agreements (SLAs) that define expected performance in terms of delay, jitter, loss and throughput. The core (or DS Domain), based on the tag on the packet (and without regard to what flow it belongs), will forward it accordingly. So instead of the more complex per-flow QoS, there is aggregate QoS. Each tag type is mapped unto a specific Per-Hop Behaviour (PHB) implemented by the router. The Assured Forwarding (AF) is a standardized PHB which consists of four classes and three dropping probabilities within each class[1]. If the actual sending rate of the user is below the minimum assured rate, packets are marked green; whereas if above the

[1]The Type-of-Service (TOS) field in the IP header is used to indicate the PHB classes and drop precedences [20]

minimum assured rate, either yellow or red [6]. A scheduler allocates bandwidth among the classes, whereas AQM enforces priorities within the class, so that during times of long-term congestion, green packets get the lowest drop rates, and red the highest. According to [12] there is intra-class fairness which depends on the AQM and inter-class fairness with depends on the scheduler. It should be noted that QoS based on this type of aggregation only provides statistical QoS assurance [12].

7.8.3 The role of IPA in Internet QoS

Because each flow that is allowed into the Intserv network is guaranteed their requested QoS, there appears to be very little need for intricate congestion control such as the IPA algorithm.

For the Diffserv architecture (which is evolving), the attempt is to push complexity to the edge nodes and simplify the core. The AQM scheme for DiffServ will not differentiate between packets that fall under the same class although they may belong to different flows. Also, for DiffServ the aggregation is not spatial[2], but is based solely on classification of packets at the edge. These two ideas differ from the IPA paradigm. If IPA is applied on a per-flow basis, then per-flow state must be maintained at the nodes. If IPA is applied on a per-aggregate basis, the aggregation is spatial, and not abstract as that used by DiffServ.

[2]By spatial aggregation flows sharing a common order of nodes can be treated as one major flow.

CHAPTER VIII

CONCLUSION

Algorithms for computing the IPA gradient estimators for loss volume and queue workload were derived for the following cases

1. A single-stage queue with instantaneous, additive loss-feedback,
2. A single-stage queue with instantaneous, additive loss-feedback and an unresponsive competing flow,
3. A single-stage queue with delayed, additive loss-feedback, and
4. A multi-stage tandem network of m queues with instantaneous, additive loss-feedback,

For all cases, the IPA gradient estimators were derived with the control parameter, θ, being the buffer-limits of the queue(s). For the single-stage case and the multi-stage case with instantaneous, additive loss-feedback, the IPA gradient estimators for when the control parameter, θ, is the loss-feedback constant, were also derived.

To verify the accuracy of the IPA algorithms, sensitivity analyses and optimizations were performed using Matlab. In Matlab, bit-level fluidization is nearly realized excepting the euler-step-size limitation arising from the numerical computation of the differential equations.

The main method of sensitivity analysis used was to run simulations for a long period (i.e., $T = 240$ seconds) for three sets of four adjacent values of θ and compute the IPA derivative in each case. Then using the first-order approximation of the Taylor series expansion as in Eq. (30) the error was calculated. This process was then repeated for three different seed values for the random number generator. It was found that the error seldom exceeded 3.5%.

For the optimization, the Stochastic Approximation (SA) algorithm was employed with the step-size being of the form $a_0 i^{-\rho}$ where a_0 is a pre-defined constant, i is the iteration number, and $0.5 < \rho < 1$. Two sets of optimizations were executed. For the first type,

each iteration used the same seed for the random number generator, (i.e., ω). In this case, if the IPA derivatives were themselves accurate, the algorithm will converge to θ^* where θ^* is a minima of the cost function $J(\theta;T)$ evaluated using the same sample-path ω. For the second type, each iteration was independent of the others, i.e. a different random seed for the random generator. For each set of optimizations, three initial values of the control parameter, θ, were used. It was found that, provided that the cost function was a convex one and that the initial value of θ was a reasonable distance from the minima, there was convergence. For the deterministic setting, it converged to the minima of the cost function. For the random setting there was convergence close to the minima.

Using the Georgia Tech Network Simulator (GTNetS), the effectiveness of these estimators in the more realistic packetized domain was demonstrated. The same methodologies for the sensitivity analysis and optimization were followed in this regime. We found that the sensitivity error tended to be larger than in the fluid-flow setting. For the single-stage cases with instantaneous and delayed loss feedback, the maginitude of the percentage error in the IPA loss-derivative never exceeded 7%. There were three occasions in all when it exceeded 6%. The queue workload-derivative only once marginally exceeded 4%. For both cases, there was convergence close to the minima when stochastic optimization was performed.

Finally, we discussed issues that should be considered if these IPA algorithms were to be implemented as part of an AQM scheme. Based on the requirements of the IPA algorithm, a virtual-queue structure at the output queue of a router was proposed for the IPA software process. This, however, still requires a sizeable amount of state information mainly due to the interdependence of queue-events among the queues. We suggested some ways to reduce this complexity. The implementation of rate-control at the source was also discussed, together with the choice of parameters, such as the time horizon, T, for the IPA computation and the performance weights for loss and queue workload. We also took a brief look at how IPA can fit into the two main proposals so far for Internet QoS, namely IntServ and DiffServ.

8.1 Future Work

This work can be extended to the multi-stage tandem network of m queues with *delayed* additive feedback. In deriving the IPA derivatives for this case, one not only has to consider the effect of events occurring at other queues but also the effect of these events when delayed. The ultimate aim of this research would be to determine a formulation for a tandem network of queues that encapsulates general forms of loss-feedback. Complementary to this would be source algorithms that more effectively co-exist with IPA-controlled queues.

Another path of research in this area would be to determine theoretically the true relationship between the stochastic fluid model (SFM) and its discrete (i.e., packetized) counterpart, since the IPA derivatives (derived in the SFM regime), performs differently in both.

Practical ways of actually implementing the IPA algorithm in real communication networks can also be explored. Extensive simulation studies may also be required to determine the effect of the new algorithms and their parameter choices on actual network performance and fairness.

APPENDIX A

SINGLE-STAGE CASE WITH COMPETING FLOW: DERIVATION OF α_1

During a full-period:

$$\alpha_1 = \sigma_1 - c\gamma\left(\frac{\alpha_1}{\alpha_1+\sigma_2}\right) \tag{240}$$

$$\gamma = \sigma_2 + \alpha_1 - \beta \tag{241}$$

Substitute Eq. (241) into Eq. (240)

$$\begin{aligned}
\alpha_1 &= \sigma_1 - c(\sigma_2+\alpha_1-\beta)\left(\frac{\alpha_1}{\alpha_1+\sigma_2}\right) \\
&\Rightarrow \alpha_1(\alpha_1+\sigma_2) = \sigma_1(\alpha_1+\sigma_2) - c(\sigma_2+\alpha_1-\beta)\alpha_1 \\
&\Rightarrow \alpha_1^2+\alpha_1\sigma_2 = \sigma_1\alpha_1+\sigma_1\sigma_2 - c\alpha_1\sigma_2 - c\alpha_1^2 + c\alpha_1\beta \\
&\Rightarrow (1+c)\alpha_1^2 + (\sigma_2-\sigma_1+c\sigma_2-c\beta)\alpha_1 - \sigma_1\sigma_2 = 0 \\
&\Rightarrow (1+c)\alpha_1^2 + ((1+c)\sigma_2-(\sigma_1+c\beta))\alpha_1 - \sigma_1\sigma_2 = 0 \\
&\Rightarrow \alpha_1 = \frac{-((1+c)\sigma_2-(\sigma_1+c\beta)) \pm \sqrt{((1+c)\sigma_2-(\sigma_1+c\beta))^2+4(1+c)\sigma_1\sigma_2}}{2(1+c)} \\
&\Rightarrow \alpha_1 = -\frac{\sigma_2(t)}{2} + \frac{\sigma_1(t)+c\beta(t)}{2(1+c)} \pm \sqrt{\left(\frac{\sigma_2(t)}{2}-\frac{\sigma_1(t)+c\beta(t)}{2(1+c)}\right)^2+\sigma_1(t)\sigma_2(t)}
\end{aligned}$$

Now, for all $\sigma_1(t), \sigma_2(t), \beta(t)$ and c:

$$\sqrt{\left(\frac{\sigma_2(t)}{2}-\frac{\sigma_1(t)+c\beta(t)}{2(1+c)}\right)^2+\sigma_1(t)\sigma_2(t)} \;>\; \left|\frac{\sigma_2(t)}{2}-\frac{\sigma_1(t)+c\beta(t)}{2(1+c)}\right|$$

so that

$$\begin{aligned}
-\frac{\sigma_2(t)}{2} + \frac{\sigma_1(t)+c\beta(t)}{2(1+c)} - \sqrt{\left(\frac{\sigma_2(t)}{2}-\frac{\sigma_1(t)+c\beta(t)}{2(1+c)}\right)^2+\sigma_1(t)\sigma_2(t)} &< 0 \\
-\frac{\sigma_2(t)}{2} + \frac{\sigma_1(t)+c\beta(t)}{2(1+c)} + \sqrt{\left(\frac{\sigma_2(t)}{2}-\frac{\sigma_1(t)+c\beta(t)}{2(1+c)}\right)^2+\sigma_1(t)\sigma_2(t)} &> 0
\end{aligned}$$

Therefore

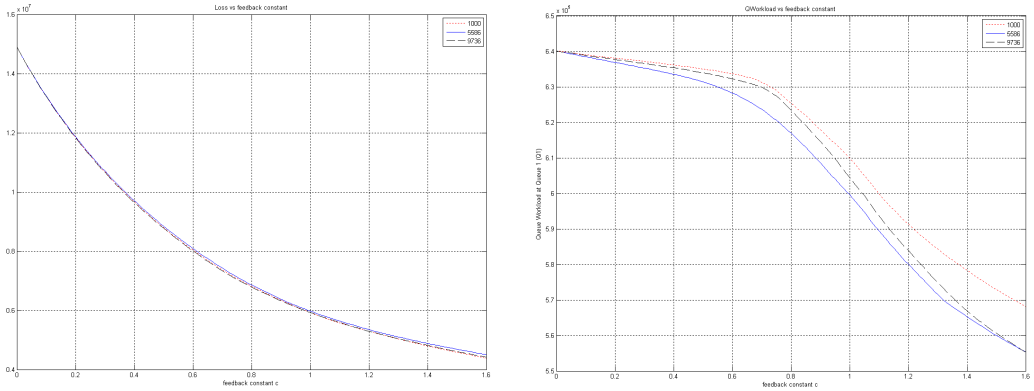
$$\Rightarrow \alpha_1 \quad = \quad -\frac{\sigma_2(t)}{2} + \frac{\sigma_1(t) + c\beta(t)}{2(1+c)} + \sqrt{\left(\frac{\sigma_2(t)}{2} - \frac{\sigma_1(t) + c\beta(t)}{2(1+c)}\right)^2 + \sigma_1(t)\sigma_2(t)}$$

Denote

$$f(\sigma_1(t), \sigma_2(t), \beta(t), c) \equiv -\frac{\sigma_2(t)}{2} + \frac{\sigma_1(t) + c\beta(t)}{2(1+c)} + \sqrt{\left(\frac{\sigma_2(t)}{2} - \frac{\sigma_1(t) + c\beta(t)}{2(1+c)}\right)^2 + \sigma_1(t)\sigma_2(t)}$$

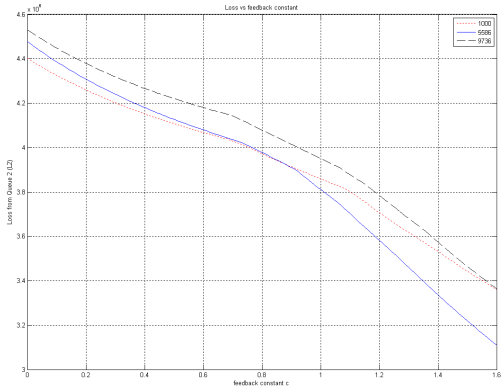
APPENDIX B

MULTI-STAGE CASE: PLOTS OF LOSS, QUEUE WORKLOAD WITH RESPECT TO THE LOSS FEEDBACK CONSTANT, C

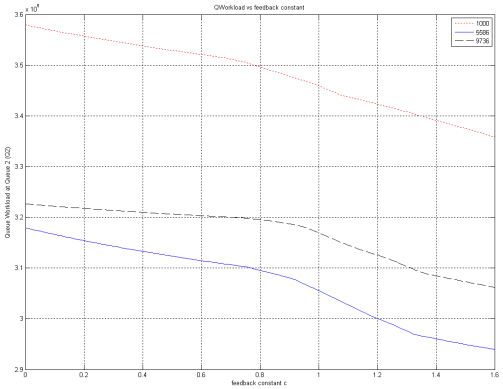


(a) Loss volume (bits) at Q_1

(b) Queue workload (bit-seconds) at Q_1



(c) Loss volume (bits) at Q_2



(d) Queue workload (bit-seconds) at Q_2

Figure 27: Loss and queue workload with respect to the loss feedback constant, c - tandem case, seed=1000

REFERENCES

[1] Agarwal, M., Gupta, R., and Kargaonkar, V., "Link utilization based AQM and its performance," in *GLOBECOM '04. IEEE Global Telecommunications Conference*, vol. 2, (Dallas, TX, USA), pp. 713–18, 2004.

[2] Anjum, F. and Tassiulas, L., "Fair bandwidth sharing among adaptive and non-adaptive flows in the Internet," in *IEEE INFOCOM '99. Conference on Computer Communications. Proceedings. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. The Future is Now*, vol. 3, (New York, NY, USA), pp. 1412–20, 1999.

[3] Arpaci, M. and Copeland, J., "An adaptive queue management method for congestion avoidance in TCP/IP networks," in *Globecom '00 - IEEE. Global Telecommunications Conference. Conference Record*, vol. 1, (San Francisco, CA, USA), pp. 309–15, 2000.

[4] Athuraliya, S., Low, S., Li, V., and Yin, Q., "REM: active queue management," *IEEE Network*, vol. 15, no. 3, pp. 48–53, 2001.

[5] Azadivar, F., "Simulation optimization methodologies," in *WSC'99. 1999 Winter Simulation Conference Proceedings. 'Simulation - A Bridge to the Future'*, vol. 1, (Phoenix, AZ, USA), pp. 93–100, 1999.

[6] Bowen, E., Jeffries, C., Kencl, L., Kind, A., and Pletka, R., "Bandwidth allocation for non-responsive flows with active queue management," in *2002 International Zurich Seminar on Broadband Communications Access - Transmission - Networking*, (Zurich, Switzerland), pp. 13–1, 2002.

[7] Brandauer, C., Iannaccone, G., Diot, C., Ziegler, T., Fdida, S., and May, M., "Comparison of tail drop and active queue management performance for bulk-data and Web-like internet traffic," in *Proceedings. Sixth IEEE Symposium on Computers and Communications*, (Hammamet, Tunisia), pp. 122–9, 2001.

[8] Bremaud, P. and Lasgouttes, J.-M., "Stationary IPA estimates for nonsmooth G/G/1/∞ functionals via palm inversion and level-crossing analysis," *Discrete Event Dynamic Systems: Theory & Applications*, vol. 3, no. 4, pp. 347–74, 1993.

[9] Bremaud, P., Malhame, R., and Massoulie, L., "A manufacturing system with general stationary failure process: stability and IPA of hedging control policies," *IEEE Transactions on Automatic Control*, vol. 42, no. 2, pp. 155–70, 1997.

[10] Brennan, R. and Rogers, P., "Stochastic optimization applied to a manufacturing system operation problem," in *1995 Winter Simulation Conference Proceedings*, (Arlington, VA, USA), pp. 857–64, 1995.

[11] Cai, L., Shen, X., Pan, J., and Mark, J., "Performance analysis of TCP-friendly AIMD algorithms for multimedia applications," *IEEE Transactions on Multimedia*, vol. 7, no. 2, pp. 339–55, 2005.

[12] Cartas, R., Orozco, J., Incera, J., and Ros, D., "A fairness study of the adaptive RIO active queue management algorithm," *Proceedings of the Fifth Mexican International Conference in Computer Science*, pp. 57–63, 2004.

[13] Cassandras, C., Sun, G., and Panayiotou, C., "Stochastic fluid models for control and optimization of systems with quality of service requirements," in *Proceedings of the 40th IEEE Conference on Decision and Control*, vol. 2, (Orlando, FL, USA), pp. 1917–22, 2001.

[14] Cassandras, C., Sun, G., Panayiotou, C., and Wardi, Y., "Perturbation analysis and control of two-class stochastic fluid models for communication networks," *IEEE Transactions on Automatic Control*, vol. 48, no. 5, pp. 770–82, 2003.

[15] Cen, S., Walpole, J., and Pu, C., "Flow and congestion control for internet media streaming applications," in *Proceedings of the SPIE - The International Society for Optical Engineering*, vol. 3310, (San Jose, CA, USA), pp. 250–64, 1997.

[16] Chatranon, G., Labrador, M., and Banerjee, S., "BLACK: detection and preferential dropping of high bandwidth unresponsive flows," in *2003 IEEE International Conference on Communications*, vol. 1, (Anchorage, AK, USA), pp. 664–8, 2003.

[17] Cheng, D., "On the design of a tandem queue with blocking: modeling, analysis, and gradient estimation," *Naval Research Logistics*, vol. 41, no. 6, pp. 759–70, 1994.

[18] Chong, E. and Ramadge, P., "Convergence of recursive optimization algorithms using infinitesimal perturbation analysis estimates," *Discrete Event Dynamic Systems: Theory & Applications*, vol. 1, no. 4, pp. 339–72, 1992.

[19] Chung, J. and Claypool, M., "Analysis of active queue management," in *Proceedings Second IEEE International Symposium on Network Computing and Applications. NCA 2003*, (Cambridge, MA, USA), pp. 359–66, 2003.

[20] Cnoddder, S. D., Pauwels, K., and Elloumi, O., "A rate based RED mechanism," in *Proceedings of the 10th ACM International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*, (Chapel Hill, NC, USA), 2000.

[21] Feng, W., Kandlur, D., Saha, D., and Shin, K., "Stochastic fair blue: a queue management algorithm for enforcing fairness," in *Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society*, vol. 3, (Anchorage, AK, USA), pp. 1520–9, 2001.

[22] Feng, W., Shin, K., Kandlur, D., and Saha, D., "The BLUE active queue management algorithms," *IEEE/ACM Transactions on Networking*, vol. 10, no. 4, pp. 513–28, 2002.

[23] Feng, W., Kapadia, A., and Thulasidasan, S., "GREEN: proactive queue management over a best-effort network," in *GLOBECOM'02 - IEEE Global Telecommunications Conference. Conference Record*, vol. 2, (Taipei, Taiwan), pp. 1774–8, 2002.

[24] Floyd, S., Gummadi, R., and Shenker, S., "Adaptive red: An algorithm for increasing the robustness of RED," 2001.

[25] Floyd, S., Handley, M., and Padhye, J., "A comparison of equation-based and AIMD congestion control." http://www.aciri.org/tfrc/., February 2000.

[26] Floyd, S. and Jacobson, V., "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397–413, 1993.

[27] Floyd, S., Handley, M., Padhye, J., and Widmer, J., "Equation-based congestion control for unicast applications," in *SIGCOMM 2000*, (Stockholm, Sweden), pp. 43–56, August 2000.

[28] Fu, M. and Hu, J.-Q., "Sample path properties of the G/D/m queue," *European Journal of Operational Research*, vol. 65, no. 2, pp. 270–3, 1993.

[29] Fu, M., Hu, J.-Q., and Nagi, R., "Comparison of gradient estimation techniques for queues with non-identical servers," *Computers & Operations Research*, vol. 22, no. 7, pp. 715–29, 1995.

[30] Fu, M. C. and Ho, Y.-C., "Using perturbation analysis for gradient estimation, averaging and updating in a stochastic approximation algorithm," in *WSC '88: Proceedings of the 20th conference on Winter simulation*, (New York, NY, USA), pp. 509–517, ACM Press, 1988.

[31] Glasserman, P., "Stationary waiting time derivatives," *Queueing Systems Theory and Applications*, vol. 12, no. 3-4, pp. 369–90, 1992.

[32] Glynn, P. W., "Optimization of stochastic systems," in *WSC '86: Proceedings of the 18th conference on Winter simulation*, (New York, NY, USA), pp. 52–59, ACM Press, 1986.

[33] Glynn, P., "Likelihood ratio derivative estimators for stochastic systems," in *1989 Winter Simulation Conference Proceedings*, (Washington, DC, USA), pp. 374–80, 1989.

[34] Glynn, P., "Pathwise convexity and its relation to convergence of time-average derivatives," *Management Science*, vol. 38, no. 9, pp. 1360–6, 1992.

[35] Handley, M., Pahdye, J., Floyd, S., and Widmer, J., "TCP friendly rate control (TFRC): Protocol specification." Request for Comments: 3448, Standards Track, Internet Engineering Task Force, January 2003.

[36] Heidergott, B., "Infinitesimal perturbation analysis for queueing networks with general service time distributions," *Queueing Systems, Theory and Applications*, vol. 31, no. 1-2, pp. 43–58, 1999.

[37] Ho, Y.-C., "Perturbation analysis; concepts and algorithms," in *1992 Winter Simulation Conference Proceedings*, pp. 231–40, 1992.

[38] Hollot, C., Misra, V., Towsley, D., and Gong, W.-B., "On designing improved controllers for AQM routers supporting TCP flows," in *Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society*, vol. 3, (Anchorage, AK, USA), pp. 1726–34, 2001.

[39] Huang, Y., Xu, G., and Huang, L., "Adaptive AIMD rate control for smooth multimedia streaming," in *Proceedings. 12th International Conference on Computer Communications and Networks*, (Dallas, TX, USA), pp. 171–7, 2003.

[40] Jiang, K., Wang, X., and Xi, Y., "Nonlinear analysis of RED - a comparative study," in *Proceedings of the 2004 American Control Conference*, vol. 4, pp. 2960–5, 2004.

[41] Kamra, A., Kapila, S., Khurana, V., Yadav, V., Saran, H., Juneja, S., and Shorey, R., "SFED: A rate control based active queue management discipline," Tech. Rep. 00A018, IBM India Research Laboratory, 2000.

[42] Kelly, F., Maulloo, A., and Tan, D., "Rate control for communication networks: shadow prices, proportional fairness and stability," *Journal of the Operational Research Society*, vol. 49, no. 3, pp. 237–52, 1998.

[43] Keshav, S. and Sharma, R., "Issues and trends in router design," *IEEE Communications Magazine*, vol. 36, no. 5, pp. 144 – 51, 1998.

[44] Kim, K. B., Tang, A., and Low, S., "Design of AQM in supporting TCP based on the well-known AIMD model," in *GLOBECOM '03. IEEE Global Telecommunications Conference*, vol. 6, (San Francisco, CA, USA), pp. 3226–30, 2003.

[45] Kompella, R. R. and Varghese, G., "Reduced state fair queuing for edge and core routers," in *Proceedings of the International Workshop on Network and Operating System Support for Digital Audio and Video*, (Cork, Ireland), pp. 100 – 105, 2004.

[46] Koo, J., Song, B., Chung, K., Lee, H., and Kahng, H., "MRED: a new approach to random early detection," in *Proceedings 15th International Conference on Information Networking*, (Beppu City, Oita, Japan), pp. 347–52, 2001.

[47] Kumaran, K. and Mitra, D., "Performance and fluid simulations of a novel shared buffer management system," in *Proceedings. IEEE INFOCOM '98, the Conference on Computer Communications. Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies. Gateway to the 21st Century*, vol. 3, (San Francisco, CA, USA), pp. 1449–61, 1998.

[48] Kunniyur, S. and Srikant, R., "Analysis and design of an adaptive virtual queue (AVQ) algorithm for active queue management," *Computer Communication Review*, vol. 31, no. 4, pp. 123–34, 2001.

[49] Kunniyur, S. and Srikant, R., "End-to-end congestion control schemes: utility functions, random losses and ECN marks," *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, pp. 689–702, 2003.

[50] Kunniyur, S. and Srikant, R., "An adaptive virtual queue (AVQ) algorithm for active queue management," *IEEE/ACM Transactions on Networking*, vol. 12, no. 2, pp. 286–99, 2004.

[51] Labovitz, C., Malan, G. R., and Jahanian, F., "Internet routing instability," *IEEE/ACM Transactions on Networking*, vol. 6, no. 5, pp. 515–528, 1998.

[52] L'Ecuyer, P., "A unified view of the IPA, SF, and LR gradient estimation techniques," *Management Science*, vol. 36, no. 11, pp. 1364–83, 1990.

[53] L'Ecuyer, P., "An overview of derivative estimation," in *1991 Winter Simulation Conference Proceedings*, (Phoenix, AZ, USA), pp. 207–17, 1991.

[54] L'Ecuyer, P., "Convergence rates for steady-state derivative estimators," *Annals of Operations Research*, vol. 39, no. 1-4, pp. 121–36, 1993.

[55] L'Ecuyer, P. and Glynn, P., "Stochastic optimization by simulation: convergence proofs for the gi/g/1 queue in steady-state," *Management Science*, vol. 40, no. 11, pp. 1562–78, 1994.

[56] Lee, K.-W., Puri, R., Kim, T., Ramchandran, K., and Bharghavan, V., "An integrated source coding and congestion control framework for video streaming in the Internet," in *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 2, (Tel Aviv, Israel), pp. 747–56, 2000.

[57] Li, N., de Veciana, G., Park, S., Borrego, M., and qi Li, S., "Minimizing queue variance using randomized deterministic marking," in *GLOBECOM'01. IEEE Global Telecommunications Conference*, vol. 4, (San Antonio, TX, USA), pp. 2368–72, 2001.

[58] Lin, D. and Morris, R., "Dynamics of random early detection," *Computer Communication Review*, vol. 27, no. 4, pp. 127–37, 1997.

[59] Long, C., Guan, X., Zhao, B., and Yang, J., "The Yellow active queue management algorithm," *Computer Networks*, vol. 47, no. 4, pp. 525–50, 2005.

[60] Low, S. H. and Lapsley, D. E., "Optimization flow control — I: basic algorithm and convergence," *IEEE/ACM Transactions on Networking*, vol. 7, no. 6, pp. 861–874, 1999.

[61] Melamed, B., Pan, S., and Wardi, Y., "Hybrid discrete-continuous fluid-flow simulation," in *Proceedings of the SPIE - The International Society for Optical Engineering*, vol. 4526, (Denver, CO, USA), pp. 263–70, 2001.

[62] Miyoshi, N., "Smoothed perturbation analysis for stationary single-server queues with multiple customer classes," *Discrete Event Dynamic Systems: Theory & Applications*, vol. 7, no. 3, pp. 275–93, 1997.

[63] Miyoshi, N., "Weak stationary solution of a G/G/1/∞ queue controlled by IPA-based sa with constant stepsize," in *Proceedings of the 37th IEEE Conference on Decision and Control*, vol. 2, (Tampa, FL, USA), pp. 1716–21, 1998.

[64] Nosrati, S. and Nikravesh, S., "Perturbation analysis for loss volume and buffer workload in tandem two-class stochastic fluid models for communication networks," in *2004 5th Asian Control Conference*, vol. 2, (Melbourne, Vic., Australia), pp. 701–7, 2004.

[65] Ott, T., Lakshman, T., and Wong, L., "SRED: stabilized red," in *IEEE INFOCOM '99. Conference on Computer Communications. Proceedings. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. The Future is Now*, vol. 3, (New York, NY, USA), pp. 1346–55, 1999.

[66] Padhye, J., Firoiu, V., Towsley, D., and Kurose, J., "Modeling TCP Reno performance: a simple model and its empirical validation," *IEEE/ACM Transactions on Networking*, vol. 8, no. 2, pp. 133–45, 2000.

[67] Pan, J. and Cassandras, C., "Infinitesimal perturbation analysis of a queueing system with bursty traffic," *Discrete Event Dynamic Systems: Theory & Applications*, vol. 4, no. 4, pp. 325–58, 1994.

[68] Panayiotou, C., "Infinitesimal perturbation analysis for a single stochastic fluid model node with a class of feedback controlled traffic," in *Proceedings of the 2004 American Control Conference*, vol. 3, (Boston, MA, USA), pp. 2308–13, 2004.

[69] Panayiotou, C., "On-line resource sharing in communication networks using infinitesimal perturbation analysis of stochastic fluid models," in *2004 43rd IEEE Conference on Decision and Control (CDC)*, vol. 1, (Nassau, Bahamas), pp. 563–8, 2004.

[70] Park, E.-C., Lim, H., Park, K.-J., and Choi, C.-H., "Analysis and design of the virtual rate control algorithm for stabilizing queues in TCP networks," *Computer Networks*, vol. 44, no. 1, pp. 17–41, 2004.

[71] Phirke, V., Claypool, M., and Kinicki, R., "RED-worcester - traffic sensitive active queue management," in *Proceedings 10th IEEE International Conference on Network Protocols*, (Paris, France), pp. 194–5, 2002.

[72] Rejaie, R., Handley, M., and Estrin, D., "RAP: An end-to-end rate-based congestion control mechanism for realtime streams in the internet," in *IEEE INFOCOM '99. Conference on Computer Communications. Proceedings. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. The Future is Now*, vol. 3, (New York, NY, USA), pp. 1337–45, 1999.

[73] Riley, G. F., "The Georgia Tech Network Simulator," in *MoMeTools '03: Proceedings of the ACM SIGCOMM workshop on Models, methods and tools for reproducible network research*, (New York, NY, USA), pp. 5–12, ACM Press, 2003.

[74] Rubinstein, R. Y. and Shapiro, A., *Discrete Event Systems: Sensitivity Analysis and Stochastic Optimization by the Score Function Method*. Wiley Series in Probability and Mathematical Statistics, John Wiley & Sons, 1993.

[75] Ryu, S., Rump, C., and Qiao, C., "Advances in active queue management (AQM) based TCP congestion control," *Telecommunication Systems - Modeling, Analysis, Design and Management*, vol. 25, no. 3-4, pp. 317–51, 2004.

[76] Spall, J. C., *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*. Discrete Mathematics and Optimization, Wiley-Interscience, 2003.

[77] Sun, G., Cassandras, C., Wardi, Y., and Panayiotou, C., "Perturbation analysis of stochastic flow networks," in *42nd IEEE International Conference on Decision and Control*, vol. 5, (Maui, HI, USA), pp. 4831–6, 2003.

[78] Sun, G., Cassandras, C., Wardi, Y., Panayiotou, C., and Riley, G., "Perturbation analysis and optimization of stochastic flow networks," *IEEE Transactions on Automatic Control*, vol. 49, no. 12, pp. 2143–59, 2004.

[79] Sun, J., Ko, K.-T., Chen, G., Chan, S., and Zukerman, M., "PD-RED: to improve the performance of RED," *IEEE Communications Letters*, vol. 7, no. 8, pp. 406–8, 2003.

[80] Suri, R. and Fu, B.-R., "On using continuous flow lines to model discrete production lines," *Discrete Event Dynamic Systems: Theory & Applications*, vol. 4, no. 2, pp. 129–69, 1994.

[81] Swisher, J. R., Hyden, P. D., Jacobson, S. H., and Schruben, L. W., "Simulation optimization: a survey of simulation optimization techniques and procedures," in *WSC '00: Proceedings of the 32nd conference on Winter simulation*, (San Diego, CA, USA), pp. 119–128, Society for Computer Simulation International, 2000.

[82] Van der Merwe, J., Sen, S., and Kalmanek, C., "Streaming video traffic: Characterization and network impact," in *Proceedings of the 7th International Workshop on Web Content Caching and Distribution, Boulder, CO, USA*, aug 2002.

[83] Wang, Y., Claypool, M., and Zuo, Z., "An empirical study of RealVideo performance across the internet," in *Proceedings of the First ACM SIGCOMM InternetMeasurement Workshop. UMW 2001*, (San Francisco, CA, USA), pp. 295–309, 2001.

[84] Wardi, Y. and Melamed, B., "Estimating nonparametric IPA derivatives of loss functions in tandem fluid models," in *Proceedings of the 40th IEEE Conference on Decision and Control*, vol. 5, (Orlando, FL, USA), pp. 4517–22, 2001.

[85] Wardi, Y., Melamed, B., Cassandras, C., and Panaytou, C., "Online IPA gradient estimators in stochastic continuous fluid models," *Journal of Optimization Theory and Applications*, vol. 115, no. 2, pp. 369–405, 2002.

[86] Wardi, Y. and Riley, G., "Ipa for spillover volume in a fluid queue with retransmissions," in *2004 43rd IEEE Conference on Decision and Control (CDC)*, vol. 4, (Nassau, Bahamas), pp. 3756–61, 2004.

[87] Wardi, Y. and Riley, G., "IPA for loss volume and buffer workload in tandem SFM networks," in *Proceedings Sixth International Workshop on Discrete Event Systems*, (Zaragoza, Spain), pp. 393–8, 2002.

[88] Wu, T., Xu, H.-B., and Tian, S.-L., "End to end congestion control and active queue management," in *Proceedings of the 2003 International Conference on Machine Learning and Cybernetics*, vol. 2, (Xi'an, China), pp. 946–50, 2003.

[89] Xu, H.-B., Wu, T., Zhou, X.-J., and Zhu, Y., "IP network control and AQM," in *Proceedings of 2004 International Conference on Machine Learning and Cybernetics*, vol. 1, (Shanghai, China), pp. 500–4, 2004.

[90] Yan, J., Katrinis, K., May, M., and Plattner, B., "Media- and TCP-friendly congestion control for scalable video streams," *IEEE Transactions on Multimedia*, vol. 8, no. 2, pp. 196–206, 2006.

[91] Yanfie, F., Fengyuan, R., and Chuang, L., "Design a PID controller for active queue management," in *Proceedings of the Eighth IEEE Symposium on Computers and Communications. ISCC 2003*, vol. 2, (Kemer-Antalya, Turkey), pp. 985–90, 2003.

[92] Yu, H. and Cassandras, C., "Perturbation analysis of feedback-controlled stochastic flow systems," in *42nd IEEE International Conference on Decision and Control*, vol. 6, (Maui, HI, USA), pp. 6277–82, 2003.

[93] Yu, H. and Cassandras, C., "Multiplicative feedback control in communication networks using stochastic flow models," in *2004 43rd IEEE Conference on Decision and Control (CDC)*, vol. 1, (Nassau, Bahamas), pp. 557–62, 2004.

[94] Yu, H. and Cassandras, C., "Perturbation analysis of feedback-controlled stochastic flow systems," *IEEE Transactions on Automatic Control*, vol. 49, no. 8, pp. 1317–32, 2004.

[95] Yu, H. and Cassandras, C., "Perturbation analysis of stochastic flow systems with multiplicative feedback," in *Proceedings of the 2004 American Control Conference*, vol. 6, (Boston, MA, USA), pp. 5734–9, 2004.

[96] Zhu, C., Yang, O., Aweya, J., Ouellette, M., and Montuno, D., "A comparison of active queue management algorithms using the OPNET modeler," *IEEE Communications Magazine*, vol. 40, no. 6, pp. 158–67, 2002.

[97] Zhu, R., Teng, H., and Fu, J., "A predictive PID controller for AQM router supporting TCP with ECN," in *APCC/MDMC '04. The 2004 Joint Conference of the 10th Asia-Pacific Conference on Communications and the 5th International Symposium on Multi-Dimensional Mobile Communications Proceeding*, vol. 1, (Beijing, China), pp. 356–60, 2004.